Information Entropy–Based Decision Making in Optimization

Tobias Christian Schmidt

Information Entropy–Based Decision Making in Optimization

Dissertation
zur
Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

dem

Fachbereich Physik
der Philipps-Universität Marburg

vorgelegt von

Tobias Christian Schmidt

aus

Mainz

Marburg/Lahn, 2009

Vom Fachbereich Physik der Philipps-Universität Marburg als Dissertation angenommen am

Erstgutachter:	Prof. Dr. Harald Ries
Zweitgutachter:	Prof. Dr. Reinhard Noack

Tag der mündlichen Prüfung:

Zusammenfassung

In der vorliegenden Arbeit werden neue Methoden zur Optimierung von stochastischen Funktionen vorgestellt. Die Methoden wurden im Hinblick auf die Optimierung von nichtabbildenden Optiken entwickelt, deren Güte in Simulationen durch Strahlverfolgung nach der Monte-Carlo-Methode evaluiert werden muss, lassen sich jedoch auf andere Anwendungen übertragen. Zum besseren Verständnis der nachfolgenden Ausführungen über Ziel und Inhalt der Arbeit werden zunächst wichtige Begriffe erläutert. Die Elemente des Definitionsbereiches einer stochastischen Funktion werden als Konfigurationen bezeichnet. Eine stochastische Funktion ist eine Funktion, deren Funktionswerte bei gegebenen Argumenten nicht direkt berechnet werden, sondern nur mithilfe von Zufallsexperimenten abgeschätzt werden können. Während es bei einer deterministischen Funktion möglich ist, Funktionswerte genau zu bestimmen, können bei einer stochastischen Funktion lediglich auf Ergebnissen von Zufallsexperimenten beruhende Wahrscheinlichkeitsverteilungen für Funktionswerte aufgestellt werden. Es wird vorausgesetzt, dass die Wahrscheinlichkeitsverteilung für den Funktionswert einer Konfiguration gegen den tatsächlichen Funktionswert konvergiert, wenn die Anzahl der Zufallsexperimente für diese Konfiguration gegen unendlich geht.

Ziel der vorliegenden Arbeit ist es, die Informationsentropie für Entscheidungen, die im Verlauf von Optimierungen stochastischer Funktionen getroffen werden, nutzbar zu machen, und auf diese Weise sehr effiziente Entscheidungen zu treffen. Effizienz bedeutet in diesem Zusammenhang, dass möglichst viel Information pro Aufwand gewonnen wird. Je höher die Effizienz eines Optimierungsalgorithmus' ist, desto bessere Optimierungsergebnisse können bei vorgegebenem Aufwand erwartet werden. Das Vorgehen orientiert sich an zwei Prinzipien: Erstens wird die Anzahl der Zufallsexperimente für jede Konfiguration dem Bedarf angepasst, denn ein Zuviel an Zufallsexperimenten bedeutet eine Verschwendung von Aufwand, zu wenige Zufallsexperimente dagegen verursachen ein schlechtes Signal-Rausch-Verhältnis, wodurch gute Funktionswerte nicht als solche erkannt werden. Zweitens wird für jede im Lauf der Optimierung zu treffende Entscheidung

die gesamte bisher erhobene Menge von Daten über die zu optimierende Funktion genutzt. Zur Umsetzung dieser beiden Prinzipien wird folgende Methode eingesetzt: Mit dem Konzept der Informationsentropie wird der Informationsgehalt der Daten, die während der Optimierung gesammelt werden, berechnet. Es werden auf diesem Informationsmaß beruhende Entscheidungskriterien formuliert, mit deren Hilfe im Lauf der Optimierung die Anzahlen der Zufallsexperimente, die für die Konfigurationen durchgeführt werden, dem Bedarf angepasst werden. Für jede zur Auswahl stehende Option wird der erwartete Informationsgewinn berechnet, dann wird die Option mit dem größten erwarteten Informationsgewinn gewählt.

In den Kapiteln 2 bis 4 werden, dieser Methode folgend, drei verschiedene Optimierungsstrategien entwickelt und getestet. Jede dieser Strategien arbeitet mit einer anderen Klasse von Optimierungsaufgaben.

In Kapitel 2 wird die Informationsentropie eingesetzt, um auf möglichst effiziente Weise Informationen über Ort und Wert des globalen Maximums einer stochastischen Funktion zu gewinnen. Der in diesem Kapitel vorgestellte Algorithmus ist für Funktionen mit endlichem Wertebereich geeignet.

In Kapitel 3 wird ein Algorithmus entwickelt, dessen Zweck es ist, mit größtmöglicher Effizienz Informationen über den Ort des globalen Maximums einer stochastischen Funktion zu gewinnen. Es wird nicht angestrebt, Information über den Wert des Maximums zu erhalten, jedoch fällt Information über den Funktionswert des Maximums als Nebenprodukt an. Dieser Algorithmus arbeitet ebenfalls nur mit Funktionen mit endlichem Wertebereich.

Das Kapitel 4 stellt einen Algorithmus vor, der ebenfalls auf der Informationsentropie beruht und der für stochastische Funktionen mit kontinuierlichem Wertebereich geeignet ist. Neben dem in Kapitel 3 eingeführten Entropiekriterium wird noch ein weiteres benutzt das entscheidet, wann bereits bekannte Konfigurationen und wann neue Konfigurationen ausgewertetet werden. Es handelt sich also um eine Erweiterung des in Kapitel 3 entwickelten Algorithmus'.

Eng mit der Optimierung stochastischer Funktionen verwandt sind die als „Ranking and Selection“ bekannten Methoden. Unter „Ranking and Selection“ versteht man Methoden, die den Zweck haben aus einer vorgegebenen Menge von Alternativen eine kleine Teilmenge auszuwählen, die mehrere gute Alternativen enthält. Im Kapitel 5 wird gezeigt, dass das Konzept der Informationsentropie auch für „Ranking and Selection“ verwendet werden kann.

Die in den Kapiteln 2, 3 und 5 beschriebenen Algorithmen wurden in Computerexperimenten getestet und mit einem Algorithmus, der die Informationsentropie nicht verwendet, verglichen. Es zeigte sich, dass die neuentwickelten, auf der Informationsentropie beruhenden Algorithmen die gestellten Optimierungsaufgaben mit höherer Effizienz lösten als der Vergleichsal-

gorithmus. Der in Kapitel 4 beschriebene Algorithmus wurde getestet, indem mit diesem Algorithmus eine nichtabbildende Optik konstruiert wurde.

In der vorliegenden Arbeit wird eine neue Verbindung zwischen dem Gebiet der Optimierung stochastischer Funktionen und der Informationstheorie geschaffen. Auf ganz andere Weise verbinden „Cross-Entropy Method“ und „Entropy Optimization“ Optimierung und Informationstheorie (vgl. [30] und [13]).

Die geplante Anwendung der in dieser Arbeit entwickelten Strategien ist die Optimierung von nichtabbildenden Optiken. In vielen Fällen muss bei einer solchen Optimierung die Güte der Optik mit Strahlverfolgung nach der Monte-Carlo-Methode ermittelt werden. Der Rechenaufwand dieses Verfahrens ist sehr hoch. Die in dieser Arbeit entwickelten Strategien dienen dazu, das Optimum mit so wenigen Strahlen wie möglich zu finden, um den Rechenaufwand so gering wie möglich zu halten. Die durchgeführten Computerexperimente zeigen, dass der Rechenaufwand tatsächlich relativ gering ist.

Statement of Originality

Chapter 2 was written together with Harald Ries and Wolfgang Spirkl and published in *Physical Review* [35]. All other parts are original work of the author. This thesis has not been submitted, either in whole or in part, for a degree at this or any other university or institution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, a connection between information theory and optimization is developed. Information entropy, the fundamental concept of information theory, is employed for the optimization of stochastic functions. By this means, optimization becomes very efficient. The basic idea is to base decisions during an optimization on a criterion derived from the concept of information entropy. According to this principle, three methods for the optimization of stochastic functions and one method for Ranking and Selection are developed. A function is stochastic if its function values cannot be calculated straightforwardly, but probability distributions for function values can be derived from the results of random experiments instead.

Optimizations of stochastic functions are applied to the design of illumination optics, construction of aerodynamic shapes, transportation planning, buffer allocation, portfolio optimization, and many other fields. For all applications, a highly efficient optimization is desirable. Efficiency is the ratio of gain of information concerning the optimum to invested effort. The higher the efficiency, the more complex the systems that can be optimized. Other methods linking concepts of information theory to optimization are the Cross-Entropy Method [30] and Entropy Optimization [13].

1.1 Overview: Optimization of Stochastic Functions

An algorithm is called a randomized search method if the generation of random numbers is part of the algorithm and decisions are based on these random numbers. In contrast, a method for the optimization of stochastic functions is one that has the purpose of optimizing functions with function values that are accessed via random experiments. A method for the optimization

of stochastic functions can be a randomized search method, but does not need to be. Optimization methods can further be separated into methods for local optimization and methods for global optimization. Methods that deal with functions with discrete domains are called combinatorial optimization methods.

What follows is a review of the most important established methods for the optimization of stochastic functions. The elements of a domain will be termed configurations. Current state-of-the-art local optimization techniques for stochastic functions are the Stochastic Approximation Method, Implicit Filtering, Evolution Strategy, Evolutionary Gradient Search, Direct Pattern Search, and Multi-Directional Search.

The Stochastic Approximation Method resembles the gradient descent algorithm, but uses stochastic approximations of the gradient of the loss function instead of the gradient itself. The stochastic approximations for the gradient are based on only two function evaluations per iteration in the Simultaneous Pertubation Stochastic Approximation Method. An overview of this method is given in [37].

Implicit Filtering is also derived from gradient descent, but needs more function evaluations to estimate the gradient. The size of the test steps is is adapted so as to reduce the influence of noise. This is a filtering of the loss function [22].

The Evolution Strategy uses the principles of biological evolution (mutation, selection, recombination) for local searches. It has the advantages that it does not need derivatives or difference quotients and that the computational effort grows only slowly with an increasing number of optimization parameters. It is designed for functions with continuous domains and is based on very few assumptions concerning the test functions, e.g., the test functions do not need to resemble quadratic forms in the vicinity of the optima. Consequently, its order of convergence can be outperformed by problem-specific algorithms. Originally designed for deterministic functions, it proved to be very robust against disturbance by noise and, because of that, well-suited to local optimizations of stochastic functions. The Evolution Strategy is explained in [28]. Its application to stochastic functions is analyzed in [5]. Evolutionary Gradient Search combines the principles of evolution with gradient descent [33].

Direct Pattern Search explores the neighborhood of a center point by deterministic test steps and chooses a new center point according to the result. The Direct Pattern Search is described in [21].

Multi-Directional Search resembles the well known Nelder-Mead Amoeba Algorithm, but is more robust [39]. A review of local optimization methods for stochastic functions is given in [3].

Methods for the global optimization of stochastic functions are Brute-Force Search, Monte Carlo Search, the Localized Random Search, Simulated Annealing, Threshold Accepting, the Great Deluge Algorithm, the Multi-Restart meta-algorithm and the Cross-Entropy Method.

Brute-Force Search involves evaluating all possible solutions and choosing the best one. It is only possible if the domain is small enough to evaluate all solutions in the time available. When a Brute-Force Search is applied to a stochastic problem, the number of random experiments per configuration has to be specified by the user.

The Monte Carlo Search selects configurations at random from the domain and evaluates them.

The Localized Random Search selects configurations from the domain according to a normal distribution centered on the best configuration found so far.

Simulated Annealing is an optimization algorithm inspired by the tendency of slow cooling materials to reach states with low energy, i.e., to come close to a global minimum of energy. The current base point in the search space is updated in each iteration so that downhill steps are more likely to be accepted than uphill steps. The probability that an uphill step is taken is controlled by a parameter labelled temperature (T), in analogy to the cooling process. The lower the T, the lower the probability for uphill steps. The temperature T is lowered in the course of the optimization according to a schedule (annealing schedule), which must be provided by the user, because it should be problem-specific. The uphill steps have the purpose of preventing the algorithm from being trapped by local minima. Still, because of the preference for downhill moves, regions with lower function values are expected to be sampled with higher density than regions with higher function values. Thus, Simulated Annealing combines local and global search; compare [24]. Article [14] shows that Simulated Annealing and other Markov Chain–based algorithms can deal with noise under certain conditions. A variant of the Simulated Annealing is Stochastic Annealing [6].

The Threshold Accepting Algorithm is similar to Simulated Annealing, but of simpler structure [12]. Also related to Simulated Annealing is the Great Deluge Algorithm [10]. Both Threshold Accepting and Great Deluge Algorithm often perform better than Simulated Annealing. An intuitive introduction to the Great Deluge Algorithm is given in [11].

The Multi-Restart Method is a simple, yet powerful meta-algorithm. It consists of a series of local searches. Every time a local optimum has been found, a new local search procedure is started with a randomly chosen starting point.

The Cross-Entropy Method was originally designed for rare event sim-

ulation but can also be used for optimization. The Cross-Entropy is used to measure the divergence between the distribution according to which the search space is sampled and an estimation of the ideal sampling distribution. This estimation is based on the information gained from previously sampled configurations. In an iterative process, the current distribution is updated until a stopping criterion is met. The optimum is estimated from the samples taken from the last distribution [30]. Applications of the Cross-Entropy Method to stochastic problems are [9] and [2]. Another approach using entropy for optimization is the Entropy Optimization Method [13].

In Entropy Optimization, the functions that are to be evaluated are entropies, whereas the strategies proposed in this work use the calculation of entropies as part of the algorithms, not of the test functions.

The main focus of the optimization strategies mentioned above is on how to choose from the domain the configurations that are to be evaluated next. Nevertheless, research has been done on how to determine the number of random experiments per configuration during a global search. In the following, important research results concerning this question are briefly described. A review of research on how many reevaluations Simulated Annealing and related methods require is given in [7].

The SANE algorithm (proposed in [7]) is a modification of the Simulated Annealing method. It is designed for the optimization of stochastic functions. In each iteration, a new configuration is chosen in the neighborhood of the current base configuration. With a certain probability p, the new configuration becomes the base configuration for the next iteration. Otherwise, the current base configuration remains the base configuration. The value p should have is derived from the theory of Simulated Annealing, depending on the current temperature and the function values. The algorithm performs random experiments for the configurations, estimates their performances according to the results of the random experiments, and chooses the new configuration as the new base configuration if its estimate is better than that of the previous base configuration. The crucial point is that the number of reevaluations is chosen so that p has the value it should have according to the theory. This gives a criterion for the number of reevaluations. Another variant of Simulated Annealing for stochastic objective functions is [1].

Article [19] derives sample schedules for the Monte Carlo Search method for stochastic functions. The schedule determines how the number of random experiments per configuration should be increased in the course of a random search for the global optimum of a stochastic function. First, it proves that the process will converge on the global optimum if the sample size is increased at a certain rate. Second, an implementation is proposed that ensures that the number of reevaluations will not increase impractically

fast. It employs t-tests to check whether the sample size is large enough. If the values to be compared are not statistically different according to the t-test, then the sample size is increased. In addition to that, the sample size is increased from time to time independently of the results of the t-tests to ensure theoretical convergence. This Variable-Sample Method is combined with Simulated Annealing in [18].

Book [30] provides schemes for adapting the number of samples in the course of global optimizations.

Stochastic Optimization is closely linked to Ranking and Selection and Sequential Sampling. Important publications concerning Ranking and Selection and Sequential Sampling include [17], [34], [26], [23], [16], and [20].

1.2 Motivation and Concept

A main question in the optimization of stochastic functions is how to determine the number of random experiments for each configuration. If the number of reevaluations is not matched exactly to the demand, the efficiency of the optimization suffers. When more reevaluations than necessary are taken, effort is obviously wasted. When the number of reevaluations is too small, wrong decisions are taken due to a lack of information. Previous work on the topic is described above. A variety of approaches dealing with that question has been proposed. Some of the approaches give only rough estimates for the number of reevaluations, others need to be supplied with problem-specific parameters by the user. Many algorithms do not store all gathered data. This is a drawback when the same configuration is visited several times and the data is deleted in between.

This dissertation gives a more general and unified answer to the question of how to determine the optimal number of random experiments for each configuration. The key point is to employ the concept of information entropy for the global optimization of stochastic functions. Information entropy is used to measure information. This gives a criterion for decisions. In each iteration, from the alternatives at hand, the one with the largest expected information gain is chosen. By this means, the algorithm can decide on the number of random experiments for each configuration. According to this principle, three optimization strategies are proposed.

The strategy developed in chapter 2 is termed Information Entropy Strategy (IES). Chapter 2 was published as an article [35]. (Chapter 2 differs slightly from [35], e.g., Table 2.3 is somewhat shorter than the corresponding table in [35].) The IES uses the concept of information entropy to gain information concerning the location and the value of the global optimum of

a stochastic function with high efficiency. It is suited to functions with finite domains.

Chapter 3 proposes the Projection Information Entropy Strategy (PIES). It is designed to gain information concerning the location of the global optimum of a stochastic function with a finite domain with optimal efficiency. Information concerning the value of the optimum is not desired, but is gained as a byproduct.

Chapter 4 presents an algorithm for the optimization of stochastic functions with continuous domain, the Continuous Information Entropy Strategy (CIES). The CIES is an extension of the PIES proposed in chapter 3. In addition to the entropy-based criterion utilized by the PIES, it makes use of a second criterion, which is also based on information entropy. This additional criterion allows the balancing of reevaluation and the evaluation of new configurations.

Chapter 5 shows that the main idea of these optimization methods can be utilized for Ranking and Selection as well as for optimization. Contrary to optimization procedures, Ranking and Selection methods do not seek to find an optimal solution, but have the purpose to find a small subset of the domain that contains several good solutions.

Some equations that are introduced in chapter 2 are repeated in the following chapters so that the chapters can be read independently from each other. As an example, the developed strategies deal with Bernoulli experiments. The strategies can be generalized to other types of random experiments. The main idea of this work is very general. It can be used in any process gathering information via random experiments to make the most efficient decisions. This includes simulation and rare-event simulation as well as optimization. The only requirements are that probability distributions can be derived from the data gathered so far that represent the remaining uncertainty and that for each random experiment the probability distribution of the possible results is known.

1.3 Implementation

The developed optimization methods were implemented by the author with Wolfram Mathematica Software. For the method described in chapter 3 Wolfram Workbench was used in addition to Mathematica, and for the method described in chapter 4, the Eclipse Ganymede software was used in addition to Mathematica.

1.4 Computer Experiments

In this work, the term experiments always refers to computer experiments. Empirical evidence means always evidence based on computer experiments.

Chapter 2

The Information Entropy Strategy

2.1 Introduction

The mathematical task of optimization is linked to thermodynamics and statistical physics in more than one way. The issue of global versus local optima is addressed by simulated annealing, see [32] and [4]. The entire optimization algorithm can be viewed as a finite time thermodynamic process in which numerical efficiency can be expressed as thermodynamical optimality, compare [38] and [31]. In this contribution we use an information entropy approach to quantify the information gained in optimization.

We propose a method to optimize stochastic functions that is based on information entropy. By stochastic function we mean a function that cannot be evaluated precisely, but to which the algorithm has only indirect access, e.g., via a Monte Carlo type experiment. Thus one can only derive a probability distribution for the stochastic function, the error of which decreases with computational effort.

The stochastic function can be described by a scheme for how to get an approximation of the merit function value from the results of the random experiments and a set of pairs (a_i, b_i), where the a_i are the configurations and where every b_i is an instruction about how to conduct a random experiment.

As an example, we choose stochastic functions whose b_i are Bernoulli experiments and whose domains contain a finite number of elements. The probability of the Bernoulli experiment b_i yielding a positive result is given by the value of the stochastic merit function g_i for configuration a_i. The task is to find the maximum of the g_i with respect to value and location: $g_{\mathrm{opt}} = g(a_{\mathrm{opt}})$. The search for the optimum should only proceed via Bernoulli

experiments.

2.2 Information Entropy Strategy

2.2.1 The Concept

The key task of any optimization algorithm is to decide at which location to evaluate the objective function next, based on past evaluations. For a stochastic function the algorithm should additionally specify the computational effort to be invested (or alternatively the precision sought). The strategy we propose in this contribution is based on maximizing the expected information gained in each step. For this we use the term 'Information Entropy Strategy'. The Information Entropy Strategy is specified so as to optimize as efficiently as possible.

What is meant by efficiency in this context? Efficiency is the ratio of gain to invested effort. We measure effort by the number of Bernoulli trials performed. The measure of gain is defined as follows. What we aspire to know are the location and the function value of the maximum. We do not seek to know the function value at other locations. Consequently we introduce the probability density function for the optimum $p_{\mathrm{opt}}(g, i)$, which expresses the probability density that the optimum occurs with configuration i and has the value g. We refer to p_{opt} as probability distribution for the optimum. We measure the information we gain concerning the optimum of the stochastic function by the decrease in information entropy of $p_{\mathrm{opt}}(g, i)$.

Now the Information Entropy Strategy can be outlined. Imagine the next Bernoulli trial is to be done for configuration j. Then we can calculate the expectation value of the entropy change which results from this Bernoulli trial. In order to decide the configuration for which the next Bernoulli trials should be performed, the expected entropy change following an additional trial at this configuration is calculated for all configurations. We choose that configuration with the largest expected entropy drop and perform the next Bernoulli trial there. Because this entropy drop is a measure of information gain and the number of Bernoulli trials is the measure of effort, we expect the maximum possible efficiency.

How the expected entropy changes and the probability distribution for the optimum are calculated is detailed in Secs. 2.2.2–2.2.5. Because the probability distribution for the optimum depends on all Bernoulli trials completed so far, the calculation of expectation values is tedious. From Sec. 2.2.2 to Sec. 2.2.5 we derive simplified expressions for these expectation values, which can be evaluated with moderate effort.

2.2.2 Definitions

The probability density p for the value of the objective for a given configuration i to be g, given that of n_i Bernoulli trials at that configuration k_i were successful is

$$p(g,i) = p(g,n_i,k_i) = \frac{(n_i+1)!}{k_i!(n_i-k_i)!} g^{k_i}(1-g)^{n_i-k_i}. \tag{2.1}$$

The merit g is the probability that a Bernoulli trial yields a positive result. It is restricted to the interval $0 \leq g \leq 1$.

The binomial distribution is

$$P_{\text{bin}}(g,n,k) = \frac{n!}{k!(n-k)!} g^k (1-g)^{n-k}.$$

For given g and n, $P_{\text{bin}}(g,n,k)$ is the probability to get a certain value k. The normalization condition for $P_{\text{bin}}(g,n,k)$ is

$$\sum_{k=0}^{n} P_{\text{bin}}(g,n,k) = 1.$$

In our case, n and k are given. The function $p(g,i)$ is the probability density for g. The normalization condition is

$$\int_0^1 p(g,i)\, dg = 1.$$

The additional factor (n_i+1) is necessary to satisfy this condition.

The probability P_b for the value of the ith configuration to be lower than a certain value g is

$$P_b(g,i) = P_b(g,n_i,k_i) = \int_0^g p(x,i)\, dx. \tag{2.2}$$

The index b signifies 'below'.

If a total of m configurations were tested then the probability $P_a(g)$ for all values to be below g is

$$P_a(g) = \prod_{i=1}^{m} P_b(g,i). \tag{2.3}$$

The index a signifies 'all below'.

Consequently the probability distribution for the optimum $p_{\text{opt}}(g,h)$ of configuration h being the best and having an objective equal to g is

$$p_{\rm opt}(g,h) = p(g,h) \prod_{i \neq h} P_b(g,i). \tag{2.4}$$

Note that the product extends over all configurations, except configuration h.

2.2.3 Entropy and Information

The total information entropy S of the probability distribution for the optimum as given in Eq. (2.4) is according to Shannon [36]:

$$S = -\sum_{i=1}^{i=m} \int_0^1 p_{\rm opt}(g,i) \ln \left[p_{\rm opt}(g,i) \right] \, dg. \tag{2.5}$$

We base the information entropy on the probability distribution for the optimum as given in Eq. (2.4) and *not* on the probability distribution of the value of the objective as given in Eq. (2.1), because we aspire to gain information about location and value of the maximum, and not about the entire function.

We choose to re-examine that configuration for which the expected information gain is largest, i.e., for which the expectation value of the entropy after performing an additional evaluation is lowest.

Calculating the total information gain in order to evaluate which configuration yields the largest gain, i.e., which i is the most 'interesting' configuration, is numerically demanding, in particular if many configurations have been extensively examined. The total information entropy is

$$\begin{aligned} S &= -\int_0^1 \sum_{i=1}^{i=m} p_{\rm opt}(g,i) \ln \left(P_a(g) \frac{p(g,i)}{P_b(g,i)} \right) \, dg \\ &= -\int_0^1 \ln[P_a(g)] \sum_{i=1}^{i=m} p_{\rm opt}(g,i) \, dg \\ &\quad -\int_0^1 \sum_{i=1}^{i=m} p_{\rm opt}(g,i) \ln \left(\frac{p(g,i)}{P_b(g,i)} \right) \, dg. \end{aligned} \tag{2.6}$$

The first integral can be evaluated explicitly because the sum is a total differential,

$$\sum_{i=1}^{i=m} p_{\rm opt}(g,i) = \frac{{\rm d}P_a}{{\rm d}g} \tag{2.7}$$

as can be seen from Eq. (2.3) and Eq. (2.4). This reduces the first term to

$$\begin{aligned} S_I &= -\int_0^1 \ln[P_a(g)] \sum_{i=1}^{i=m} p_{\rm opt}(g,i)\, dg \\ &= -\int_{P_a=0}^{P_a=1} \ln(P_a)\, dP_a = 1. \end{aligned} \tag{2.8}$$

The second term is:

$$\begin{aligned} S_{II} &= -\int_0^1 \sum_{i=1}^{i=m} p_{\rm opt}(g,i) \ln\left(\frac{p(g,i)}{P_b(g,i)}\right) dg \\ &= -\sum_{i=1}^{i=m} \int_0^1 \left(\prod_{j\neq i} P_b(g,j)\right) p(g,i) \ln\left(\frac{p(g,i)}{P_b(g,i)}\right) dg. \end{aligned} \tag{2.9}$$

2.2.4 Expectation Value of the Entropy

When we decide to perform one Bernoulli trial for configuration j, we expect the system to have a certain entropy $\langle S\rangle^j$ afterwards. The entropy change depends on the outcome of the Bernoulli trial.

The expectation value $\langle S\rangle^j$ after one additional event for configuration j is

$$\langle S\rangle^j = \alpha_j S^{j+} + (1-\alpha_j) S^{j-}. \tag{2.10}$$

Here $\alpha_j = (k_j+1)/(n_j+2)$ is the probability of getting a positive result when re-examining the configuration j which has a record of k_j positive results out of n_j, S^{j+} is the total entropy following a successful Bernoulli trial, where n_j and k_j would both be increased by one. Consequently $1-\alpha_j = (n_j+1-k_j)/(n_j+2)$ is the probability for a negative result and S^{j-} the entropy after a negative result if only n_j is increased by one.

2.2.5 Calculating the Expected Entropy Change

With Eqs. (2.9) and (2.10), the expected change in the total entropy due to one additional Bernoulli trial for configuration j can be calculated. Equation (2.10) yields:

$$\langle \Delta S\rangle^j = \alpha_j S^{j+} + (1-\alpha_j) S^{j-} - S. \tag{2.11}$$

Later, we will use the important fact that

$$\alpha_j \frac{P_b^{j+}(g,j)}{P_b^{(0)}(g,j)} + (1-\alpha_j)\frac{P_b^{j-}(g,j)}{P_b^{(0)}(g,j)} = 1. \tag{2.12}$$

This is a consequence of the fact that a priori the expected probability distribution after a measurement is equal to the distribution before the measurement. This is a property of all probability distributions.

Now S, S^{j+}, S^{j-} are calculated using Eq. (2.9) and substituted into Eq. (2.11):

$$\begin{aligned} S &= 1 - \sum_{i=1}^{i=m} \int_0^1 \left(\prod_{j \neq i} P_b(g,j) \right) p(g,i) \ln \left(\frac{p(g,i)}{P_b(g,i)} \right) dg \\ &= 1 - \sum_{i=1}^{i=m} \int_0^1 \left(\frac{P_a(g)}{P_b(g,i)} \right) p(g,i) \ln \left(\frac{p(g,i)}{P_b(g,i)} \right) dg \\ &= 1 - \int_0^1 P_a(g) \sum_{i=1}^{i=m} \left(\frac{p(g,i)}{P_b(g,i)} \right) \ln \left(\frac{p(g,i)}{P_b(g,i)} \right) dg. \end{aligned} \tag{2.13}$$

In the following, the superscript $^{(0)}$ refers to values calculated before a new Bernoulli trial is carried out, whereas the superscript $^{j+}$ refers to a value calculated assuming a new Bernoulli trial was successful, and $^{j-}$ refers to a value assuming it was unsuccessful.

Now, how does S change when one additional measurement (Bernoulli trial) is successfully performed for configuration j? The integrand in Eq. (2.13) consists of a product and a sum. After a new measurement, one of the factors of the product changes and one of the summands of the sum. Thus, we can replace the initial terms by the new ones, which yields:

$$S^{j+} = 1 - \int_0^1 P_a^{(0)}(g) V^{j+} \left(A + T^{j+} - T^{(0)} \right) dg. \tag{2.14}$$

With the following abbreviations:

$$\begin{aligned} V^{j+} &= \frac{P_b^{j+}(g,j)}{P_b^{(0)}(g,j)}, \\ V^{j-} &= \frac{P_b^{j-}(g,j)}{P_b^{(0)}(g,j)}, \\ A &= \sum_{i=1}^{i=m} \frac{p^{(0)}(g,i)}{P_b^{(0)}(g,i)} \ln \frac{p^{(0)}(g,i)}{P_b^{(0)}(g,i)}, \end{aligned}$$

$$\begin{aligned} T^{(0)} &= \frac{p^{(0)}(g,j)}{P_b^{(0)}(g,j)} \ln \frac{p^{(0)}(g,j)}{P_b^{(0)}(g,j)}, \\ T^{j+} &= \frac{p^{j+}(g,j)}{P_b^{j+}(g,j)} \ln \frac{p^{j+}(g,j)}{P_b^{j+}(g,j)}, \\ T^{j-} &= \frac{p^{j-}(g,j)}{P_b^{j-}(g,j)} \ln \frac{p^{j-}(g,j)}{P_b^{j-}(g,j)}. \end{aligned} \tag{2.15}$$

Similarly:

$$S^{j-} = 1 - \int_0^1 P_a^{(0)}(g) V^{j-} \left(A + T^{j-} - T^{(0)} \right) \, dg. \tag{2.16}$$

And, of course,

$$S = 1 - \int_0^1 P_a^{(0)}(g) \; A \, dg. \tag{2.17}$$

Equations (2.14), (2.16), and (2.17) are substituted into Eq. (2.11). Then, we use the fact that

$$\left[\alpha_j V^{j+} + (1 - \alpha_j) V^{j-} \right] = 1. \tag{2.18}$$

See Eq. (2.12). This yields

$$\begin{aligned} \langle \Delta S \rangle^j &= -\alpha_j \left(\int_0^1 P_a^{(0)}(g) V^{j+} \left(T^{j+} - T^{(0)} \right) \, dg \right) \\ &\quad -(1 - \alpha_j) \left(\int_0^1 P_a^{(0)}(g) V^{j-} \left(T^{j-} - T^{(0)} \right) \, dg \right). \end{aligned} \tag{2.19}$$

The term A no longer shows up in the equation. The result can further be simplified using Eq. (2.18):

$$\langle \Delta S \rangle^j = \int_0^1 P_a^{(0)}(g) \left[T^{(0)} - \alpha_j V^{j+} T^{j+} - (1 - \alpha_j) V^{j-} T^{j-} \right] \, dg. \tag{2.20}$$

This is the main equation for our strategy. Every time we want to decide the configuration for which the next Bernoulli trial should be made, we evaluate $\langle \Delta S \rangle^j$ for all configurations j and conduct the Bernoulli trial where $-\langle \Delta S \rangle^j$ is largest. (The minus is because the smaller the entropy, the more knowledge one has.) In practice it is not necessary to calculate the $\langle \Delta S \rangle^j$ before every Bernoulli trial. Rather, we assume that the change in $\langle \Delta S \rangle^j$ is small when a small number of Bernoulli trials are made for a certain configuration. By 'a small number' we mean small compared with the total number

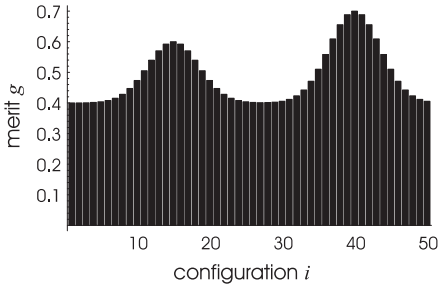

Figure 2.1: The example function.

of Bernoulli trials. Hence, we proceed as follows: we calculate the $\langle \Delta S \rangle^j$, then we make a small number of Bernoulli trials for the configuration for which $-\langle \Delta S \rangle^j$ is largest, then we recalculate the $\langle \Delta S \rangle^j$, and so on.

2.3 Test Results

2.3.1 Applications

The application we have in mind is to choose the best from a set of virtual optical systems for illumination via Monte Carlo ray tracing. This is a standard procedure in optical design. Sending a randomly chosen ray through a virtual illumination optic is a Bernoulli trial. If the ray strikes the target surface, the outcome is 'true,' otherwise it is 'false.' Hence, every one of these illumination optic systems is an instruction on how to do a Bernoulli trial, and hence can be a b_i. If the illumination systems are a discrete subset of a parameterized set, the a_i is the parameter vector which specifies the illumination system b_i. Otherwise, one can think of the a_i simply as names of the illumination systems. By stochastic optimization we aspire to find the illumination system which directs more radiation onto the target than any of the others.

2.3.2 A Naive Strategy Used for Comparison

We use a naive and simple strategy for solving the introduced optimization problem as a benchmark for the Information Entropy Strategy. The simple strategy carries out the same number of Bernoulli trials at all configurations. From the basic theorem of Monte Carlo integration, the necessary number of Bernoulli trials per configuration is calculated [27]. Finding out the probability g_i of a 'true' result for a Bernoulli trial by repeatedly performing

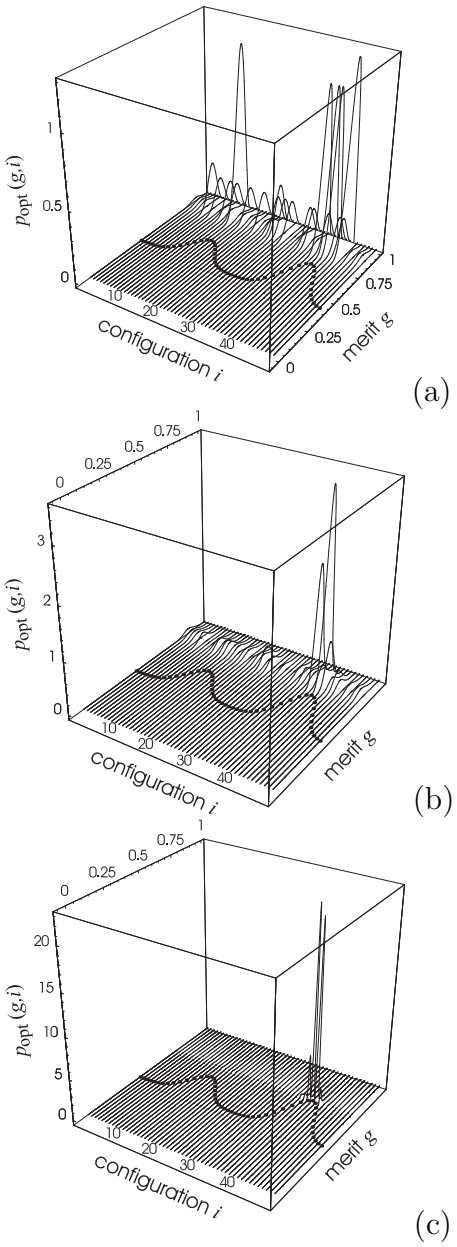

Figure 2.2: Optimization with Information Entropy Strategy. The three graphs plot the probability distribution for the optimum $p_{\rm opt}(g,i)$ for different stages of the optimization process. In each graph, the example function is shown with dots in the horizontal plane and the probability distribution for the optimum $p_{\rm opt}(g,i)$ is plotted in the vertical direction for each of the 50 configurations as a function of the merit function value. The plot in graph (a) is based on 300 Bernoulli trials, the plot in graph (b) on 500, and the plot in graph (c) on 10^4. The information entropy of the probability distribution for the optimum shown in (c) is -2.12.

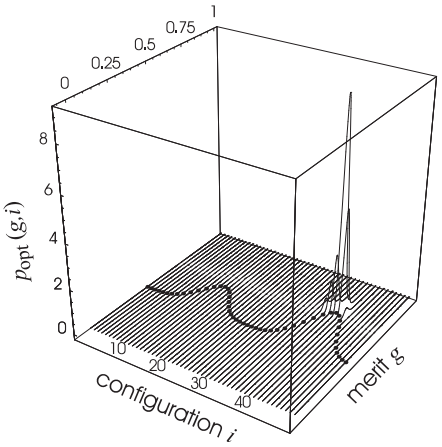

Figure 2.3: This graph shows the probability distribution for the optimum calculated from 10^4 Bernoulli trials, distributed according to the naive strategy among the configurations of the example function. The probability distribution for the optimum $p_{\mathrm{opt}}(g, i)$ is plotted for each of the 50 configurations as a function of the merit function value. In the horizontal plane the example function is shown with dots. The information entropy of the probability distribution for the optimum shown is -1.15.

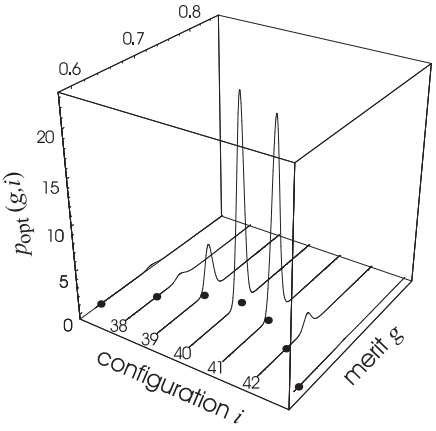

Figure 2.4: This graph is a magnified section of Fig. 2.2, graph (c). It shows the probability distribution for the optimum in the vicinity of its maximum for the Information Entropy Strategy after 10^4 Bernoulli trials. The information entropy of the probability distribution for the optimum shown is -2.12.

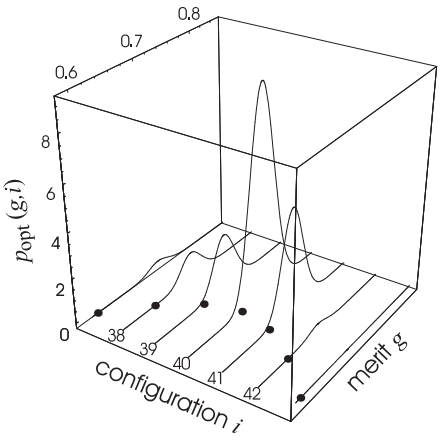

Figure 2.5: This graph is a magnified section of Fig. 2.3. The information entropy of the probability distribution for the optimum shown is -1.15.

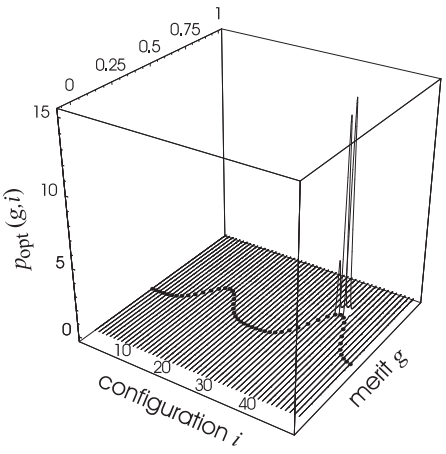

Figure 2.6: This graph shows the probability distribution for the optimum calculated from 4×10^4 Bernoulli trials, distributed according to the naive strategy among the configurations of the example function. The probability distribution for the optimum $p_{\mathrm{opt}}(g, i)$ is plotted for each of the 50 configurations as a function of the merit function value. In the horizontal plane the example function is shown with dots. The information entropy of the probability distribution for the optimum shown is -2.02.

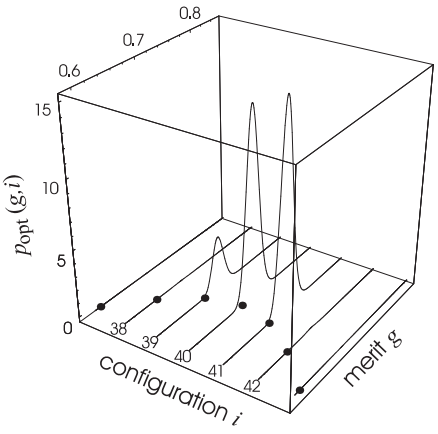

Figure 2.7: This graph is a magnified section of Fig. 2.6. It shows the details of the peak. The information entropy of the probability distribution for the optimum shown is -2.02.

Bernoulli trials is equivalent to integrating a function $f(x)$ with $x \in [0,1]$ and $f(x) = 1 | x \leq g_i$ and $f(x) = 0 | x > g_i$ with the Monte Carlo method and determining g_i from the result.

An error estimate for the integral is

$$\epsilon = V \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}.$$

Here, N is the number of randomly chosen points and V is the volume, over which the integration extends. The brackets indicate averaging:

$$\langle f \rangle \equiv \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \text{ and } \langle f^2 \rangle \equiv \frac{1}{N} \sum_{i=0}^{N-1} f^2(x_i).$$

For this error estimation see Ref. [27], chapter 7.

Since $x \in [0,1]$, $V = 1$ and because of $f \in \{0,1\}$, $f(x_i) = f^2(x_i)$, and $\langle f^2 \rangle = \langle f \rangle$, this yields

$$\epsilon = \sqrt{\langle f \rangle - \langle f \rangle^2} \sqrt{\frac{1}{N}}.$$

Because $\langle f \rangle \in [0,1]$, the maximum of $\sqrt{\langle f \rangle - \langle f \rangle^2}$ is 0.5. If $\langle f \rangle$ is not known in advance, the error estimate is

$$\epsilon_{\max} = \frac{1}{2\sqrt{N}}.$$

To get an error smaller than $\epsilon_{\max}$, the necessary number of Bernoulli trials per configuration is

$$N = \frac{1}{4\epsilon_{\max}^2}.$$

2.3.3 Test Function

We tested the Information Entropy Strategy by applying it to an example function defined for a discrete set of 50 configurations distinguished by one parameter: $a_1 = 0.02; a_2 = 0.04; ...; a_{50} = 1$. The corresponding 'true' merit function values are chosen to express two peaks of different heights.

$$g_i = 0.2 \exp\left[-\left(\frac{a_i - 0.3}{0.1}\right)^2\right] + 0.3 \exp\left[-\left(\frac{a_i - 0.8}{0.1}\right)^2\right] + 0.4,$$

with $i \in \{1; ...; 50\}$. This is called the example function, see Fig. 2.1. A Bernoulli trial for configuration a_i is made like this: a random number between 0 and 1 is generated. If it is smaller than g_i, the result is true, otherwise the result is false. These instructions are called b_i. A total of 10000 Bernoulli trials were distributed among the configurations according to the Information Entropy Strategy. We chose to make five Bernoulli trials every time the $\langle \Delta S \rangle^j$ were calculated. The graphs in Fig. 2.2 show the probability distribution for the optimum in different stages of the optimization process. In the beginning, one cannot see from the probability distribution for the optimum where the maximum lies, or what value it has, but after 10000 Bernoulli trials, the location and value of the maximum are found with good precision.

2.3.4 Performance Comparison

Figure 2.2 illustrates the evolution of the probability distribution for location and value of the optimum in the course of an optimization following the Information Entropy Strategy. Figure 2.2(a) refers to the result after a total of 300 Bernoulli trials were completed, Fig. 2.2(b) after 500 experiments, and finally Fig. 2.2(c) after 10^4 Bernoulli trials were carried out. The information entropy of the probability distribution for the optimum at this point was -2.12. Note that the probability distribution for the optimum at the beginning [Fig. 2.2(a)] shows two peaks after which it settles at the higher peak.

We have compared the Information Entropy Strategy to the naive strategy. Figure 2.3 shows the probability distribution for the optimum calculated from 10^4 Bernoulli trials, which were distributed according to the naive

strategy among the configurations of the example function. The information entropy of this probability distribution for the optimum is -1.15.

Note that after an equal number of evaluations the naive strategy correctly identifies the global maximum of the test function, however, the distribution is much broader, i.e., the maximum is identified with less precision. This is illustrated in more detail in Figs. 2.4 and 2.5 which enlarge the relevant range close to the optimum of Figs. 2.2(c) and 2.3.

For a better comparison, we allowed the naive strategy to continue until the probability distribution for the optimum roughly matched the results of Fig. 2.2(c). See Figs. 2.6 and 2.7. At this point the information entropy was -2.02. We found that a total of 4×10^4 Bernoulli trials were necessary. This illustrates the superiority of the Information Entropy Strategy.

2.3.5 Computation Time

The computation time used by the Information Entropy Strategy is split between the time needed for carrying out the Bernoulli trials and the overhead needed to evaluate the expected entropy gain in order to decide which configuration to examine next. For this decision Eq. (2.20) needs to be evaluated for each configuration. Thus the time needed is roughly proportional to the number of configurations, i.e., the size of the system.

In the examples presented in Sec. 2.3.4 the stochastic function used allowed a very fast evaluation of Bernoulli trials. Furthermore the expected entropy gain was evaluated very frequently (every five Bernoulli trials). Consequently, the overhead dominated the computation time in these examples. However, this is not to be expected in practical applications, for several reasons:

- Additional Bernoulli trials change the expected entropy less if many trials have been previously performed. Therefore the number of Bernoulli trials carried out between consecutive evaluations of the expected entropy gain should increase in the course of the optimization, eventually rendering the computation time spent for Bernoulli trials dominant.

- For practical applications the computation involved in carrying out Bernoulli trials is probably more time consuming than in the simple tests used here. In particular we envision using the Information Entropy Strategy for the design of optical illumination systems, where performance is assessed via Monte Carlo Ray tracing. In this field a Bernoulli trial would be equivalent to tracing a ray through an optical system which involves finding intersections at each optical surface. The

duration for complex systems which may involve freeform surfaces may well be over 1 ms/ray.

- We did not code the evaluation of the expected entropy gain in the most efficient way yet. For example the term $P_a^{(0)}$ in Eq. (2.20) may be evaluated recursively much faster than directly via Eq. (2.3) as currently done. It is also possible that in the course of optimization, some configurations are recognized to be so uninteresting that they need not be considered at each evaluation.

In order to compare the Information Entropy Strategy with the naive strategy in terms of computation time in a remotely realistic way with our present code, we simply used a stochastic function, which was implicitly defined via a numerical root finding, such that the time needed for the Bernoulli trials was much longer. We used this implicit test function with systems of 50, 200, and 800 configurations. The results are summarized in Tables 2.1–2.3. After an initial phase, during which the naive strategy is faster, the Information Entropy Strategy is faster in reducing the entropy of the probability distribution for the maximum. The duration of this initial phase increases with system size.

This finding is easily explained: Initially, as a priori all configurations are equal, the Information Entropy Strategy coincides with the naive strategy in the choice of where to evaluate the stochastic function. Therefore the naive strategy is superior because it has no overhead. After a rough localization of the maximum, the better choice made by the Information Entropy Strategy offsets the overhead. For larger systems a rough localization of the maximum takes longer.

2.3.6 The Special Case of Two Equally High Maxima

Up to now, we have only been concerned with stochastic functions which have exactly one global maximum. Since in practical applications the number of maxima is not known beforehand, it is important to know how the algorithm proceeds in the case of several equally high local maxima. Consequently, we tested the Information Entropy Strategy on a test function with two equally high maxima, which we call the degenerate test function.

The degenerate test function is

$$g_i = 0.25 \exp\left[-\left(\frac{a_i - 0.3}{0.1}\right)^2\right] + 0.25 \exp\left[-\left(\frac{a_i - 0.8}{0.1}\right)^2\right] + 0.4, \quad (2.21)$$

for the parameter values $a_1 = 0.02; a_2 = 0.04; ...; a_{50} = 1$.

Table 2.1: Results of the optimization of the implicit example function with 50 configurations: (a) Naive strategy, (b) Information Entropy Strategy.

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.98	0
50	-3.586	5
100	-3.939	11
150	-4.120	16
200	-4.251	22
250	-4.382	28

(a)

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.98	0
25	-0.771	4
50	-3.700	9
75	-5.198	16
100	-5.516	26

(b)

Table 2.2: Results of the optimization of the implicit example function with 200 configurations: (a) Naive strategy, (b) Information Entropy Strategy.

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.995	0
200	-3.053	22
400	-3.930	45
600	-4.087	67
800	-4.186	89
1000	-4.321	112

(a)

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.995	0
25	0.994	7
50	0.993	14
75	0.992	20
100	0.957	28
125	0.880	36
150	0.621	45
175	-0.271	56
200	-3.440	67
225	-5.138	87

(b)

Table 2.3: Results of the optimization of the implicit example function with 800 configurations: (a) Naive strategy, (b) Information Entropy Strategy.

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.999	0
1600	−2.576	205
3200	−2.710	402
4800	−3.101	599
6400	−3.708	795
8000	−3.944	992
9600	−3.960	1189
11200	−4.159	1389
12800	−4.222	1586
14400	−4.061	1787
16000	−4.197	1991

(a)

Number of Bernoulli trials in 10^3	Entropy	Computing time in minutes
0	0.999	0
100	0.999	67
200	0.998	144
300	0.998	227
400	0.998	321
500	0.997	427
600	0.994	544
700	0.928	675
800	−2.608	816
900	−3.973	1092
1000	−4.394	1525
1100	−4.724	2141

(b)

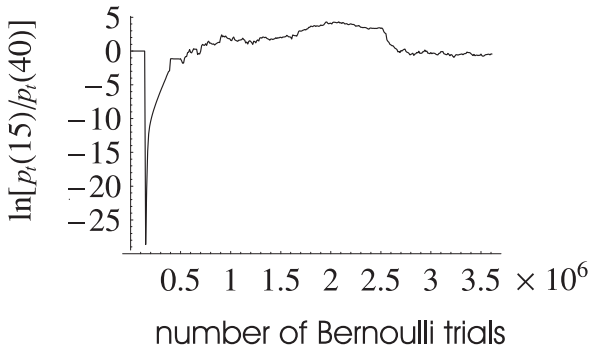

Figure 2.8: Optimization of the degenerate test function with Information Entropy Strategy. The probability $p_t(15)$ is the total probability that the global maximum is at 0.3, whereas $p_t(40)$ is the total probability that the global maximum is at 0.8. This graph shows $\ln[p_t(15)/p_t(40)]$ as a function of the number of Bernoulli trials. The total number of Bernoulli trials is 3.61×10^6.

We want to ensure that the Information Entropy Strategy identifies both maxima, and not only one of them. For each of the maxima we integrate the probability distribution for the optimum $p_{\text{opt}}(g, i)$ over the merit function value g, thus getting the total probability $p_t(i)$ that this configuration is better than all others. The probability $p_t(i)$ is a function of the number of Bernoulli trials, since the probability distribution for the optimum is a function of the number of Bernoulli trials. Both of the maxima are found if $p_t(i)$ is of the same order of magnitude for both of the maxima and small for all the other configurations. The result is shown in Figs. 2.8 and 2.9 and in Table 2.4. Two things can be learned from Figs. 2.8 and 2.9. First, one can see that for sufficiently large numbers of Bernoulli trials $p_t(15)$ and $p_t(40)$ are of the same order of magnitude and all other $p_t(i)$ are small compared to $p_t(15)$ and $p_t(40)$, since the sum of all $p_t(i)$ is one. That means that both maxima were found by the Information Entropy Strategy (see Fig. 2.10). Second, the number of Bernoulli trials should not be too small. If the calculation had been stopped after 2×10^6 Bernoulli trials, the maximum a_{40} perhaps had been overlooked (see Table 2.4).

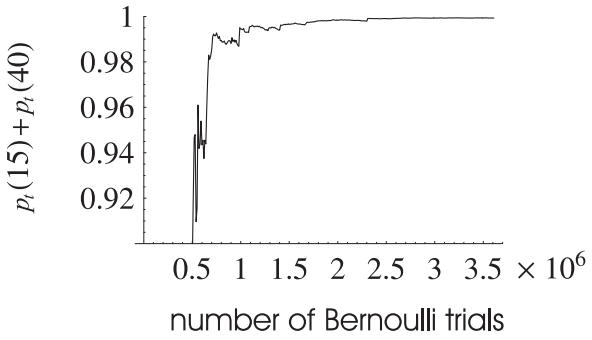

Figure 2.9: Optimization of the degenerate test function with Information Entropy Strategy. The probability $p_t(15)$ is the total probability that the global maximum is at 0.3, whereas $p_t(40)$ is the total probability that the global maximum is at 0.8. This graph shows the sum $p_t(15) + p_t(40)$ as a function of the number of Bernoulli trials. The total number of Bernoulli trials is 3.61×10^6.

Table 2.4: Optimization of the degenerate test function with Information Entropy Strategy. The table shows the total probability for the two maximal configurations for different numbers of Bernoulli trials.

Number of Bernoulli trials	$p_t(15)$	$p_t(40)$
1×10^6	0.830	0.165
2×10^6	0.982	0.016
3×10^6	0.439	0.560
3.61×10^6	0.389	0.610

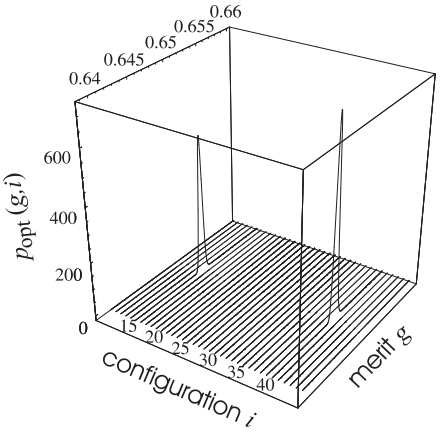

Figure 2.10: Optimization with the Information Entropy Strategy. The graph shows the probability distribution for the optimum for the degenerate test function after 3.61×10^6 Bernoulli trials. The probability distribution for the optimum is negligible outside of the section shown in the graph.

2.4 Conclusions

Information entropy appears to be a useful criterion for the optimization of stochastic functions. However, it is important in the context of optimization to base the information entropy on the probability distribution for the optimum rather then the probability distribution of the stochastic function itself.

Chapter 3

The Projection Information Entropy Strategy

3.1 Introduction

Optimization means gaining information concerning the extremum of a function. Two different sorts of information are of interest in the optimization of stochastic functions, namely information about the location of the optimum and information about the function value of the optimum. Optimizations of stochastic functions, especially simulation-based optimizations, are often used for design, e.g., optical design. When an optimization is run in order to determine design parameters, the the user is mainly interested in information concerning the location of the optimum. For this reason, in this chapter a strategy is developed with the purpose to find the location of the optimum as efficiently as possible. It is not aspired to know the value of the optimum. Still, some information about the value of the optimum is gained as a by-product of the search for the location of the optimum.

As the IES, this strategy is based on the concept of information entropy, and it is developed to optimize stochastic functions, the function values of which cannot be straightforward evaluated but can only be estimated by random experiments. Again, it is assumed that the precision of the estimation increases monotonically with increasing number of random experiments and that the estimation converges on the exact function value in the limit of an infinite number of random experiments.

This chapter is concerned with the optimization of functions on finite domains. Functions with continuous domains are treated in chapter 4. Stochastic functions that are estimated via Bernoulli experiments are chosen as an example. For each element of the domain, the probability of a Bernoulli

trial yielding a positive result is given by the function value. The optimization algorithm has no direct access to the function values, but estimates the function values from the results of the Bernoulli trials.

The strategy described in chapter 2 sought to collect information about both location and value of the maximum. If both sorts of information are sought, the variant proposed there should be used. If, however, mainly information about the location of the maximum is sought, in particular if some decision (e.g. choice of construction parameters) is to be based on the location of the maximum, the strategy proposed in the present chapter is recommended, since it is superior to the method described in chapter 2 for this type of task. The strategy described in chapter 2 is termed 'Information Entropy Strategy', abbreviated IES. The strategy proposed in this chapter is referred to as 'Projection Information Entropy Strategy' (PIES), since it uses the information entropy of a projected probability distribution. In the present chapter, the PIES is explained and compared with the IES and the naive strategy.

3.2 Application

A possible application of the PIES is the design of non-imaging optical systems. In the design of illumination optics, the performance of optical systems is determined by tracing randomly chosen light rays through virtual prototypes of the optical systems (Monte Carlo Method). Imagine the goal is to direct as much light as possible from a given light source onto a given target, and that from a set of virtual prototypes of optics that direct light from the source to the target, the best one is to be determined. In this case, the merit function is the fraction of the light which reaches the target, and each ray is a Bernoulli trial. If the ray hits the target, the result is 'true', otherwise it is 'false'. It is the purpose of the PIES to find the best optical system from the set while tracing as few rays as possible through the virtual prototypes. For illumination optics, typically several million rays are needed. The Monte Carlo method is explained in [27].

3.3 The Strategy

3.3.1 The Concept

To optimize efficiently, it is necessary to gain as much information per Bernoulli trial as possible concerning the location of the maximum. Efficiency is the ratio of gain to invested effort. The number of Bernoulli trials performed

is the measure of effort. The measure of gain is defined as follows: The probability $p_t(i)$ is the probability that configuration i is the optimum. The elements of the domain are termed configurations. An information entropy is associated with every probability distribution. The information entropy of the distribution $p_t(i)$ measures the amount of information we have about the location of the optimum. Consequently, the change of information entropy corresponds to the information gain. The measure of gain is the change of information entropy. This definition of efficiency gives a criterion for decisions. Before performing a Bernoulli trial, for each configuration the change in information entropy that is expected to result from an additional Bernoulli trial performed for this configuration is calculated. The next Bernoulli trial is performed for the configuration with the largest expected information gain.

While the measure of effort is identical to the one used in chapter 2, the measure of gain is significantly different. This provides a much higher efficiency when the user is interested in the location of the maximum alone rather than being interested in location and value of the optimum.

3.3.2 Definitions

Some of the definitions from chapter 2 are recapitulated here: The probability density of configuration i to have the merit g is $p(g, i)$. The number of Bernoulli trials performed for configuration i is n_i, and k_i of these were successful.

$$p(g, i) = p(g, n_i, k_i) = \frac{(n_i + 1)!}{k_i!(n_i - k_i)!} g^{k_i} (1 - g)^{n_i - k_i}. \tag{3.1}$$

The probability of configuration i to have a merit lower than g is $P_b(g, i)$.

$$P_b(g, i) = P_b(g, n_i, k_i) = \int_0^g p(x, i)\, dx. \tag{3.2}$$

The probability of all values to be below g is $P_a(g)$.

$$P_a(g) = \prod_{i=1}^{m} P_b(g, i). \tag{3.3}$$

Here, m is the number of configurations. The probability distribution of configuration h being the best and having an objective equal to g is $p_{\text{opt}}(g, h)$.

$$p_{\text{opt}}(g, h) = p(g, h) \prod_{i \neq h} P_b(g, i). \tag{3.4}$$

The product includes all configurations, except the configuration h. In addition, $p_t(i)$ is defined to be the probability that configuration i is the location of the optimum.

$$p_t(i) = \int_0^1 p_{\text{opt}}(g, i)\, dg. \tag{3.5}$$

3.3.3 Information Entropy

The theory of information entropy is explained in [36]. The information entropy S_p of the probability distribution of the location of the optimum is:

$$\begin{aligned} S_p &= -\sum_{i=1}^{i=m} [p_t(i) \ln p_t(i)] && (3.6) \\ &= -\sum_{i=1}^{i=m} \left[\left(\int_0^1 p_{\text{opt}}(g, i)\ dg \right) \ln \left(\int_0^1 p_{\text{opt}}(g, i)\ dg \right) \right]. && (3.7) \end{aligned}$$

Note that this equation is significantly different from the corresponding equation in chapter 2, Eq. (2.5). The index p signifies 'projected'. This index is used since S_p is based on the $p_t(i)$, which are projections of $p_{\text{opt}}(g, i)$ on the configurations. (See Eq. (3.5).) The entropy S as defined in chapter 2 (Eq. (2.5)) is now denoted S_f.

$$S_f = -\sum_{i=1}^{i=m} \int_0^1 p_{\text{opt}}(g, i) \ln \left(p_{\text{opt}}(g, i) \right)\ dg. \tag{3.8}$$

The index f signifies 'full'. This index is used since S_f is based on $p_{\text{opt}}(g, i)$, which is the full probability distribution, giving information about location and value of the maximum.

3.3.4 Expectation Value of the Entropy Change

In chapter 2, how to calculate the expectation value of the entropy change $\langle \Delta S \rangle^j$ resulting from an additional Bernoulli trial was explained. Analogously, the expected change of S_p is calculated. The expected entropy $\langle S_p \rangle^j$ after one additional Bernoulli trial for configuration j is

$$\langle S_p \rangle^j = \alpha_j S_p^{j+} + (1 - \alpha_j) S_p^{j-}. \tag{3.9}$$

Here, $\alpha_j = (k_j + 1)/(n_j + 2)$ is the probability of the additional Bernoulli trial yielding a positive result, n_j is the number of Bernoulli trials previously

performed for configuration j, and k_j is the number of positive results. The entropy following a successful Bernoulli trial is S_p^{j+} and the entropy following an unsuccessful Bernoulli trial is S_p^{j-}. Consequently, the expected value of the entropy change is:

$$\langle \Delta S_p \rangle^j = \alpha_j S_p^{j+} + (1 - \alpha_j) S_p^{j-} - S_p. \tag{3.10}$$

The most interesting configuration is the one for which $-\langle \Delta S_p \rangle^j$ is largest.

3.3.5 The Algorithm

The algorithm evaluates expected entropy changes and performs Bernoulli trials in turn. A small number of Bernoulli trials is performed for the most interesting configuration after each evaluation of the expected entropy changes. 'A small number' means a number that is small compared to the total number of Bernoulli trials. The maximum efficiency in terms of information gain per Bernoulli trial is reached if only one Bernoulli trial is performed per iteration, but then the frequent evaluation of expected entropy changes results in high computational cost.

The PIES in pseudocode

Initialization
Choose $n_{\text{total}} \in \mathbf{N}$
Choose $n_{\text{inc}} \in \mathbf{N}$ with $1 \leq n_{\text{inc}} \ll n_{\text{total}}$

Optimization
REPEAT
    Evaluate $\langle \Delta S_p \rangle^i$ for each configuration i in the domain
    Perform n_{inc} Bernoulli trials
        for the configuration with the smallest $\langle \Delta S_p \rangle^i$
UNTIL
    n_{total} Bernoulli trials have been performed
Result
Evaluate $p_t(i)$ for each configuration i in the domain
This probability distribution $p_t(i)$ is the optimization result

Notes

- The probabilities $p_t(i)$ are evaluated according to Eq. (3.5).
- The probability that configuration i is the best of the evaluated configurations is $p_t(i)$.
- The expected entropy changes $\langle \Delta S_p \rangle^i$ are evaluated according to Eq. (3.10).

3.4 Test

In order to evaluate the performance of the PIES, three different algorithms are compared by applying them to the same test problem. These are the IES, the PIES, and a naive strategy which performs the same number of Bernoulli trials for each configuration. The strategy of performing the same number of Bernoulli trials per configuration will be referred to as the naive strategy in the following. It is used as a benchmark.

3.4.1 The First Test Function

Since the test optimizations are repeated several times to reduce the noise, let us choose a simple test function in order to keep the computation time moderate. The test function has a unique maximum. The test function is

$$g_i^{(p1)} = 0.5 - 10\ a_i^2, \tag{3.11}$$

where $i \in \{1, ..., 5\}$ and the configurations are

$$(a_1, a_2, a_3, a_4, a_5) = (-0.2, -0.1, 0, 0.1, 0.2).$$

The merit function value at configuration i is $g_i^{(p1)}$. See Fig. 3.1. The superscript $^{(p1)}$ means that this is the first test function used to test the PIES.

The value of the merit function is estimated by performing Bernoulli trials. Performing a Bernoulli trial for configuration i means generating a random number between 0 and 1 and comparing it to the value $g_i^{(p1)}$ of the merit function at that configuration. If the random number is smaller than $g_i^{(p1)}$, the result is true, otherwise it is false.

3.4.2 The Test

Each of the three strategies performed 200 Bernoulli trials per iteration. The naive strategy performed 40 of the 200 Bernoulli trials at each of the five

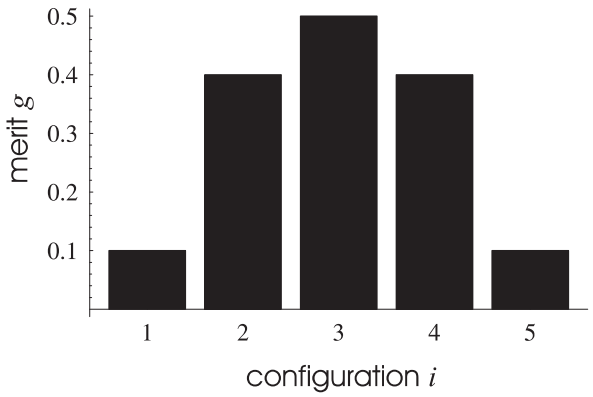

Figure 3.1: The first test function for the PIES: $g^{(p1)}$

configurations. The IES and the PIES performed the 200 Bernoulli trials for the configuration with the largest expected entropy drop, whereby they used different distributions to calculate the entropy as pointed out in Secs. 3.1–3.3. Each of the algorithms performed a total of 10^4 Bernoulli trials for one optimization, and each of the algorithms was applied 20 times to the test problem. In addition to that, as an initialization, 100 Bernoulli trials were carried out per configuration before the algorithms were started.

3.4.3 Test Results

The three strategies are compared according to two different criteria. One criterion for the performance of an optimization is the entropy S_p, calculated from the $p_t(i)$. The other criterion for the performance of an optimization is the entropy S_f, calculated from the $p_{\text{opt}}(g, h)$. Using S_p as a criterion for the performance of the strategies gives qualitatively different results than using S_f.

Fig. 3.2 shows a comparison of three optimization strategies. From Fig. 3.2 it can be seen that in terms of S_p, the PIES is the best of the three strategies, and that even the naive strategy is superior to the IES, except at the beginning of the optimization.

Fig. 3.3 shows a comparison of the same strategies with the other criterion of comparison. The data is taken from the same test optimizations as shown in Fig. 3.2. Contrary to S_p, S_f can have negative values. This is due to the fact that the $p_t(i)$ represent a discrete probability distribution, whereas $p_{\text{opt}}(g, h)$ is continuous in g. Consequently, $-\ln S_f$ is not well defined. From Fig. 3.3, it can be seen that, in terms of S_f, the IES is the best strategy and

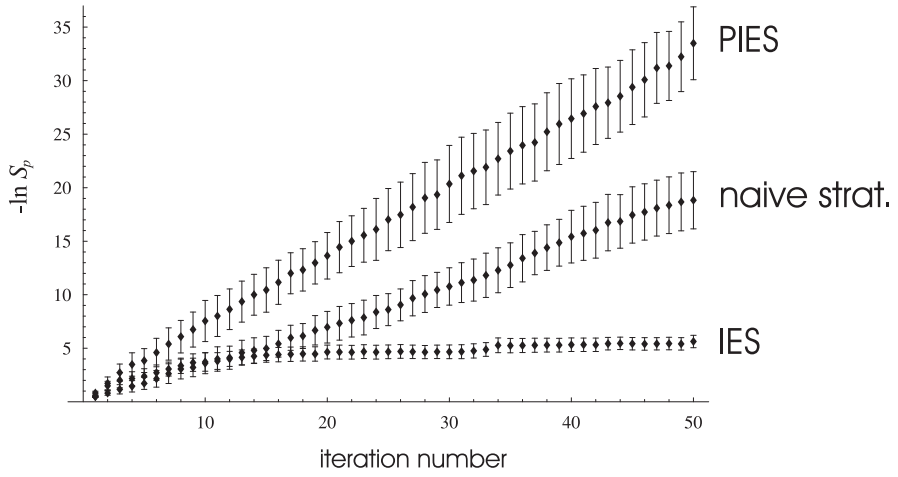

Figure 3.2: Comparison of three optimization strategies. From the bottom up: IES, naive strategy, PIES. The negative logarithm of the Entropy S_p is plotted against the iteration number. 200 Bernoulli trials were carried out per iteration. Each strategy was run 20 times, the diamonds represent the arithmetic mean of $-\ln S_p$ over the 20 test runs. The error bars mark confidence intervals in which the expectation values of $-\ln S_p$ lie with 95% probability. The test function was $g^{(p1)}$.

the PIES is better than the naive strategy.

Note that in Fig. 3.2 the best strategy is that above, whereas in Fig. 3.3 the best strategy is plotted below the two others. This is due to the negative sign in $-\ln S_p$.

Table 3.1 shows how the algorithms distributed the Bernoulli trials among the configurations and the resulting $p_t(i)$. The brackets $''\langle ...\rangle''$ denote the average over the results of 20 test runs. The results are rounded.

3.4.4 Interpretation of the Test Results

The test results are discussed in the following. First, note that the naive strategy is the best neither in Fig. 3.2 nor in Fig. 3.3. Information-entropy-based methods outperform the naive strategy for this type of test function. Second, the results show that there is a significant difference between the IES and the PIES. Third, the claim about the user's intentions and their choice of Information Entropy Strategy variant has been confirmed. A user who desires to know only the location of the maximum is interested in achieving a low S_p. Fig. 3.2 shows that, in this case, the PIES should be chosen.

Table 3.1: Results of the optimizations of test function $g^{(p1)}$. The brackets $''\langle ... \rangle''$ denote the average over the results of 20 test runs. Configuration 3 is the optimum. The result of the PIES is closest to the ideal probability distribution (0,0,1,0,0).

i	$\langle n_i \rangle$	$\langle k_i \rangle$	$\langle p_t(i) \rangle$
1	100	8	2.3×10^{-19}
2	500	210	7.9×10^{-4}
3	9700	4845	0.9991268463723
4	100	31	8.3×10^{-5}
5	100	12	1.2×10^{-15}

IES

i	$\langle n_i \rangle$	$\langle k_i \rangle$	$\langle p_t(i) \rangle$
1	2100	208	5.1×10^{-197}
2	2100	854	2.7×10^{-7}
3	2100	1054	0.9999997281587
4	2100	834	1.3×10^{-9}
5	2100	209	6.4×10^{-200}

naive strategy

i	$\langle n_i \rangle$	$\langle k_i \rangle$	$\langle p_t(i) \rangle$
1	170	15	8.1×10^{-19}
2	3410	1375	6.6×10^{-12}
3	3960	1977	0.9999999999924
4	2680	1059	9.3×10^{-13}
5	280	30	1.4×10^{-15}

PIES

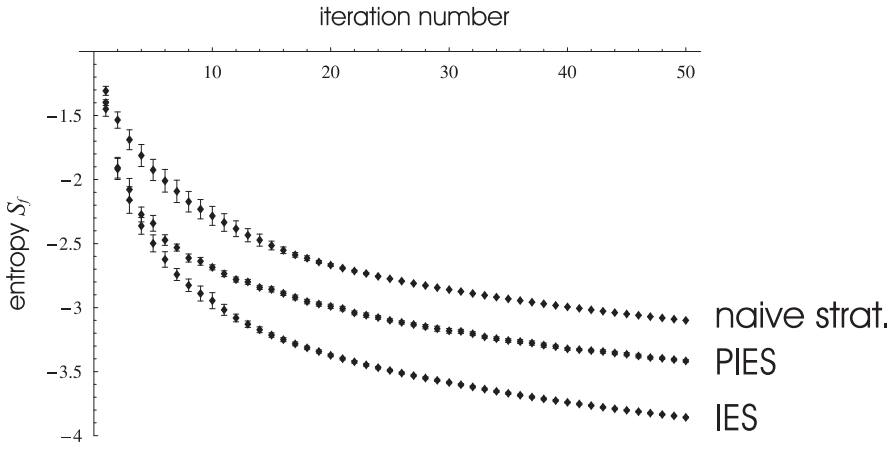

Figure 3.3: Comparison of three optimization strategies. The data refers to the same test optimizations as shown in Fig. 3.2. From the bottom up: IES, PIES, naive strategy. The entropy S_f is plotted against the iteration number. 200 Bernoulli trials were carried out per iteration. Each strategy was run 20 times, the diamonds represent the arithmetic mean of S_f over the 20 test runs. The error bars mark confidence intervals of 95% probability. The test function was $g^{(p1)}$.

However, a user who wants to know about both the location and the value of the maximum, and hence is interested in achieving a low S_f, should choose the IES, as can be seen from Fig. 3.3.

3.4.5 Important Remark

Note that in terms of S_p the IES is inferior to the naive strategy (Fig. 3.2). This means that, if the aim is to find the location of the maximum, the IES should not be used. It should be used only if the aim is to collect information about both the location and value of the optimum.

3.5 Approximations and Numeric Analysis

3.5.1 Approximations

Approximation for Expected Entropy Changes

Evaluating the expected entropy changes according to Eqs. (3.7) and (3.10) requires much computation time. For that reason, an approximation is used

to evaluate the $\langle \Delta S_p \rangle^j$. The test in Sec. 3.5.3 indicates that using this approximation leads to an efficiency (in terms of information gain per Bernoulli trial) that is almost as high as when $\langle \Delta S_p \rangle^j$ is evaluated with Eqs. (3.7) and (3.10). It is not necessary that $\langle \Delta S_p \rangle^j$ is well-approximated in every iteration, only that the approximation is able to identify configurations with high information gain. The approximation of $\langle \Delta S_p \rangle^j$ is:

$$\langle \Delta S_{ap} \rangle^j = \alpha_j \; F(p_t(j)^{j+}) + (1 - \alpha_j) \; F(p_t(j)^{j-}) - F(p_t(j)), \tag{3.12}$$

where

$$F(x) := -x \ln x - (1 - x) \ln(1 - x). \tag{3.13}$$

The approximation only works properly if for each configuration a number $n_i >> 1$ of Bernoulli trials have already been performed.

The index *ap* signifies 'approximation'. Here $p_t(j)$ is the probability that the maximum occurs with configuration j, $p_t(i)^{j+}$ is the probability that the maximum occurs with configuration i, given that the next Bernoulli experiment is performed for configuration j and gives a positive result, and $p_t(i)^{j-}$ is the probability that the maximum occurs with configuration i, given that the next Bernoulli experiment is performed for configuration j and gives a negative result. $p_t(j)^{j+}$ and $p_t(j)^{j-}$ are the special cases with $i = j$. Refer to Sec. 3.3.4 for the definition of α_j.

Series Expansion

A series expansion is used to evaluate $\langle \Delta S_{ap} \rangle^j$.
Definitions:
$\varepsilon^{j+} := p_t(j)^{j+} - p_t(j)$ and $\varepsilon^{j-} := p_t(j)^{j-} - p_t(j)$. The expansion is:

$$\begin{aligned} F(x + \varepsilon) &= -(x + \varepsilon) \; \ln(x + \varepsilon) \\ &\quad -(1 - (x + \varepsilon)) \; \ln(1 - (x + \varepsilon)) \\ &\approx ((-1 + x) \; \ln(1 - x) - x \; \ln x) \\ &\quad +(\ln(1 - x) - \ln(x)) \; \varepsilon \\ &\quad -\frac{\varepsilon^2}{2 \; x \; (1 - x)} \\ &\quad +\ldots \; . \end{aligned} \tag{3.14}$$

The small parameter is ε.

This yields:

$$\begin{aligned} \langle \Delta S_{ap} \rangle^j &= \alpha_j \; F(p_t(j) + \varepsilon^{j+}) \\ &\quad +(1-\alpha_j) \; F(p_t(j) + \varepsilon^{j-}) \\ &\quad -F(p_t(j)) \\ &\approx \alpha_j \; B_1 + (1-\alpha_j) \; B_2. \end{aligned} \tag{3.15}$$

Here B_1 and B_2 are:

$$B_1 = (\ln(1-p_t(j)) - \ln(p_t(j))) \; \varepsilon^{j+} - \frac{(\varepsilon^{j+})^2}{2 \; p_t(j) \; (1-p_t(j))} + \ldots \; .$$

$$B_2 = (\ln(1-p_t(j)) - \ln(p_t(j))) \; \varepsilon^{j-} - \frac{(\varepsilon^{j-})^2}{2 \; p_t(j) \; (1-p_t(j))} + \ldots \; .$$

The zero order terms cancel:

$$\begin{aligned} & \alpha_j \; [(-1+p_t(j)) \; \ln(1-p_t(j)) - p_t(j) \ln p_t(j)] \\ + \; & (1-\alpha_j) \; [(-1+p_t(j)) \; \ln(1-p_t(j)) - p_t(j) \; \ln p_t(j)] \\ - \; & F(p_t(j)) \\ = \; & (\alpha_j + (1-\alpha_j)) \; F(p_t(j)) - \; F(p_t(j)) \\ = \; & 0. \end{aligned}$$

Here, only the first and second order terms of the series expansion are written down. In all optimizations using this series expansion, the terms up to ε^{10} were taken into account. When $\langle \Delta S_{ap} \rangle^j$ is calculated via Eq. (3.15) the numerical error is smaller then when Eq. (3.12) is used instead. This is the case because Eq. (3.12) describes a small difference of larger numbers, which yields large relative errors. This difficulty does not occur in Eq. (3.15). Indeed, when $\varepsilon^{j+} = p_t(j)^{j+} - p_t(j)$ and $\varepsilon^{j-} = p_t(j)^{j-} - p_t(j)$ are calculated, the difference is also much smaller than minuend and subtrahend, but the ratio of difference and minuend, or respectively difference and subtrahend is larger than it is for Eq. (3.12). This yields a smaller numerical error.

3.5.2 Numeric Analysis

This subsection explains some details of the numeric analysis.

Beta Functions

The function $P_b(g, n_i, k_i)$ is related to the Euler beta function. The function $B_z(q_1, q_2)$ is the incomplete beta function.

$$B_z(q_1, q_2) = \int_0^z (\tilde{z})^{q_1-1}(1-\tilde{z})^{q_2-1}\, d\tilde{z}.$$

The Euler beta function is $B(q_1, q_2) = B_1(q_1, q_2)$.
The function $I_z(q_1, q_2)$ is the regularized incomplete beta function.

$$I_z(q_1, q_2) = \frac{B_z(q_1, q_2)}{B(q_1, q_2)}.$$

The relation

$$P_b(g, n_i, k_i) = I_g(1 + k_i, 1 + n_i - k_i)$$

can be used to evaluate $P_b(g, n_i, k_i)$.

Treatment of Probabilities Close to One

Probabilities $p_t(j)$ close to one are calculated as the difference between one and the sum of all other probabilities:

$$p_t(j) = 1 - \sum_{i,\ i\neq j} p_t(i).$$

For such probabilities, for the expression $(1 - p_t(j))$ in Eq. (3.15), the sum of all other probabilities is inserted to avoid small differences of larger numbers.

The same method is used for the evaluation of $p_t(j)^{j+}$ and $p_t(j)^{j-}$ when they are close to one.

$$p_t(j)^{j\pm} = 1 - \sum_{i,\ i\neq j} p_t(i)^{j\pm}.$$

Numerical Integration

For the evaluation of the $\langle \Delta S_{ap} \rangle^j$ the integration in Eq. (3.5) is essential. The numbers $j \in \{1, 2, ..., m\}$ are the numbers of the configurations. For all configurations j the same supporting points are used to integrate

$$p_t(j) = \int_0^1 p_{\text{opt}}(g, j)\, dg$$

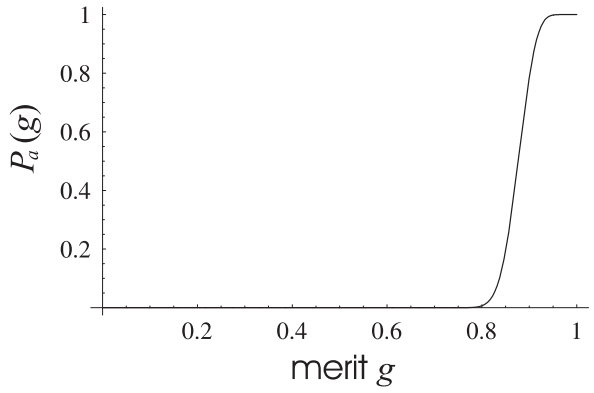

Figure 3.4: Typical form of $P_a(g)$. The probability that all configurations have a merit lower than g is plotted over g.

numerically. The $p_{\text{opt}}(g, j)$ are evaluated with the equation

$$p_{\text{opt}}(g, j) = \frac{p(g, j)}{P_b(g, j)} P_a(g), \tag{3.16}$$

which is equivalent to Eq. (3.4). The probabilities $P_a(g)$ are stored for all the supporting points. So $P_a(g)$ needs to be evaluated only once for each supporting point and can be used for all configurations j. The same supporting points are also used for the evaluation of $p_t(j)^{j\pm}$ and $p_t(i)^{j\pm}$. Due to the approximation (Eq. (3.12) and Eq. (3.15)), not all $p_t(i)^{j\pm}, i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., m\}$ have to be evaluated. It is sufficient to evaluate $p_t(j)^{j\pm}$ for all configurations and the $p_t(i)^{j\pm}, i \in (\{1, 2, ..., m\} \backslash \{j\})$ for the j with $p_t(j)^{j\pm}$ close to one. There can be at most one configuration with $p_t(j)^{j\pm}$ close to one per iteration.

Fig. (3.4) shows a typical form of the function $P_a(g)$. If the number of Bernoulli trials is large enough, there are intervals where $P_a(g)$ is close to zero or one. Those intervals contribute little to the integral of Eq. (3.16). The size of the interval in which the slope of $P_a(g)$ is large decreases when the number of Bernoulli trials is increased. In this interval, the space between the supporting points must be small, whereas a larger space can be between the supporting points in the intervals where $P_a(g)$ is approximatively constant. With this method, the numerical integration is rather precise with a moderate number of supporting points. New supporting points are chosen in each iteration.

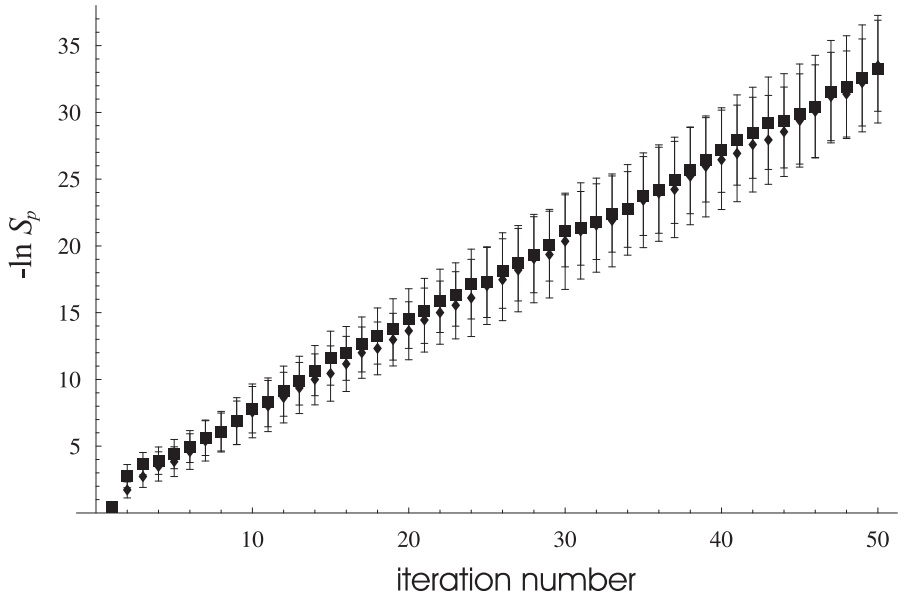

Figure 3.5: A comparison of the approximation introduced in Sec. 3.5 and the PIES without approximations. The approximation is plotted with box-shaped symbols, the strategy without approximation with diamond-shaped symbols. The plot with diamond-shaped symbols is identical to the highest one in Fig. 3.2. The negative logarithm of the entropy S_p is plotted against the iteration number. 200 Bernoulli trials were carried out per iteration. Each strategy was run 20 times, the diamonds respectively squares represent the arithmetic mean of $-\ln S_p$ over the 20 test runs. The error bars mark confidence intervals of 95% probability. The test function was $g^{(p1)}$.

3.5.3 Test of the Approximation

To test the approximations and methods of numeric analysis described in Sec. 3.5.1 and Sec. 3.5.2 empirically, the PIES, utilizing the approximations and methods of numeric analysis, was applied to the test function $g_i^{(p1)}$, which was used to compare the strategy variants in Sec. 3.4. Again, 50 iterations with 200 Bernoulli trials per iteration were made. Note that a sufficient number of Bernoulli trials per configuration is essential for the approximation to work properly. Therefore 100 Bernoulli trials per configuration were performed before the algorithm was started. As before, the optimization was repeated 20 times.

The approximation is compared to the exact calculation. The criterion of comparison is S_p. Fig. 3.5 shows that, for this example, the efficiency of the approximative algorithm is almost as high as the efficiency of the strategy

without approximation. The difference between the two is not significant since the distance between diamond-shaped and box-shaped symbols is much smaller than the error bars.

3.6 Computation Time

3.6.1 The Overhead

The computational load of the PIES is divided among two processes, namely the performing of Bernoulli trials and the evaluation of expected entropy changes. The latter is termed the overhead. While the PIES needs less Bernoulli trials than the naive strategy to give good results, the naive strategy has the advantage of having no overhead. In this section, two methods for reducing the overhead are provided and the PIES is compared to the naive strategy with respect to time consumption. These methods can be used with the IES and the CIES (chapter 4) as well.

First Overhead Reduction Method

The number of Bernoulli trials performed for a certain configuration after an evaluation of the expected entropy changes is n_{inc}. In the previous sections, n_{inc} was kept constant during an optimization. Since the number of performed Bernoulli trials increases in the course of the optimization, n_{inc} can also be increased. This can be done by performing a number n_{start} of Bernoulli trials per configuration before starting the optimization and choosing $n_{\text{inc}}(n_{\text{old}}) = \text{round}(\max\{c_1, c_2 \cdot n_{\text{old}}\})$. Here n_{old} is the total number of Bernoulli trials that have already been performed for the configuration for that the n_{inc} new Bernoulli trials shall be performed, and c_1 and c_2 are positive constants. If c_1 and c_2 are sufficiently small, then n_{inc} is small compared to the total number of Bernoulli trials and is increased during the optimization. With this method, the ratio of overhead and time for the Bernoulli trials improves in the course of the optimization until the overhead is negligible compared to the time consumed by the Bernoulli trials (compare chapter 2).

Second Overhead Reduction Method

The second method for reducing the overhead is to perform Bernoulli trials for several configurations in each iteration. Let $\delta \in]0, 1[$ be a threshold. Then evaluate in each iteration all configurations with expected entropy change between $\langle \Delta S_p \rangle^{\text{best}}$ and $\delta \cdot \langle \Delta S_p \rangle^{\text{best}}$. Here, $\langle \Delta S_p \rangle^{\text{best}}$ is the minimum of the $\langle \Delta S_p \rangle^j$ calculated in a certain iteration, i.e., the expected entropy change for

the most interesting configuration. With this method, more Bernoulli trials are performed per iteration, while the overhead is not changed. Consequently, the fraction of total computation time due to the overhead is reduced.

3.6.2 Example of Time Consumption

For practical applications it is important to know whether the PIES is efficient only in terms of the number of Bernoulli trials or whether it is also efficient in terms of time consumption. If the overhead exceeds the time that is saved by the fact that the Information Entropy Strategy needs fewer Bernoulli trials than the naive strategy, then the PIES is not useful in practice. In the following the time consumption of the PIES and the naive strategy are compared.

The function to test the time consumption is

$$g_i^{(p2)} = 0.03 \exp\left[-\left(\frac{a_i - 0.3}{0.1}\right)^2\right] + 0.045 \exp\left[-\left(\frac{a_i - 0.8}{0.1}\right)^2\right] + 0.4, \quad (3.17)$$

where $i \in \{1, 2, ...100\}$ and the configurations are

$$(a_1, a_2, ..., a_{100}) = (0.01, 0.02, ...1).$$

The merit function at configuration i is $g_i^{(p2)}$. The superscript $^{(p2)}$ means that this function is the second function used to test the PIES. The value of the merit function is estimated by performing Bernoulli trials. Performing a Bernoulli trial for configuration i means generating a random number between 0 and 1 and comparing it to the value $g_i^{(p2)}$ of the merit function at that configuration. If the random number is smaller than $g_i^{(p2)}$, the result is true, otherwise it is false.

For each of the 100 configurations, 10^4 Bernoulli trials were performed before the start of the test optimizations. The naive strategy performed 200×10^6 Bernoulli trials in addition to the initial 10^4 per configuration. The PIES performed about 100×10^6 Bernoulli trials on this test function in addition to the initial 10^4 per configuration. The approximations and methods of numeric analysis as given in Sec. 3.5 and both of the overhead reduction methods introduced in Sec. 3.6.1 were utilized. The constants c_1 and c_2 were set to

$$\begin{aligned} c_1 &= 10^5, \\ c_2 &= 0.1, \\ \delta &= 0.6. \end{aligned}$$

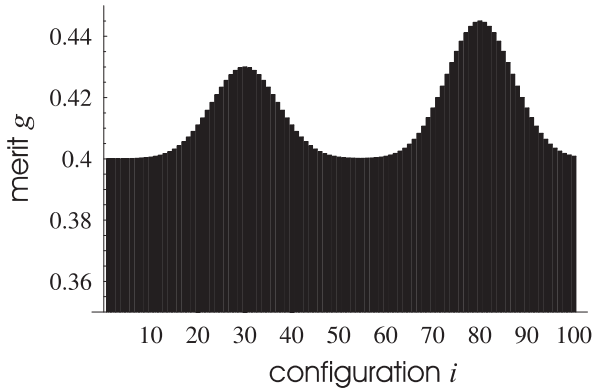

Figure 3.6: The second test function for the PIES: $g^{(p2)}$. It was used for the evaluation of the time consumption.

Table 3.2: Results of the test of the time consumption

Strategy	Overhead	Number of Bernoulli trials in 10^6	Calculated time[a] consumption for Bernoulli trials	Total time[b] consumption	Entropy S_p
naive	0	200	55.56 h	55.56 h	0.6889575
PIES	1.44 h[c]	108.5	30.14 h	31.58 h	0.0000597

[a]Calculated under the assumption that each Bernoulli trial takes 1 ms.
[b]Calculated time for Bernoulli trials plus overhead.
[c]Determined with a test run on a 2.40GHz / 512 MB RAM PC.

Refer to Sec. 3.6.1 for the definition of $c_1, c_2,$ and δ.

Performing a Bernoulli trial as described above takes a very short time. For the application to non-imaging optics, a Bernoulli trial is performed by tracing a ray through a virtual optical system. Tracing one ray takes typically one millisecond. Hence, for both the PIES and the naive strategy the time was calculated that would have been consumed if each Bernoulli trial had taken 1 millisecond. The results are shown in Table 3.2. The PIES reaches a lower entropy than the naive strategy while the computation time is shorter.

3.7 Conclusions

The Information Entropy Strategy and the Projection Information Entropy Strategy found the maximum of a stochastic function on a finite domain with high efficiency. It is crucial to choose the correct probability distribution to calculate the entropy. The correct choice of the distribution should reflect the user's intentions. A wrong choice renders the strategy less efficient than the naive strategy. Numerical methods for evaluating expected entropy changes with moderate effort are given.

Chapter 4

The Continuous Information Entropy Strategy

4.1 Introduction

This chapter proposes an algorithm that is an extension of the PIES. The PIES is an algorithm based on information entropy for the optimization of stochastic functions on finite domains (see chapter 3). In this chapter, the PIES is extended to functions on continuous domains. Because the number of possible configurations is infinite, the algorithm should gradually increase the number of examined configurations. Thus a criterion for when to add a new configuration to the set of configurations examined is added. This criterion is also based on information entropy. The extended strategy is termed CIES for Continuous Information Entropy Strategy.

What properties do the functions that the CIES optimizes have? The functions are stochastic, i.e., the function values cannot be evaluated directly, but can only be estimated from the results of random experiments. In the limit of infinitely many random experiments, the estimation error approaches zero. As an example, functions that are evaluated via Bernoulli experiments are chosen. To evaluate a configuration means to perform a Bernoulli trial for the configuration. The functions have continuous domains that can extend over many dimensions, i.e., such a domain is an open subset of the $\mathbb{R}^N$ or the closure of such a subset. The points in the domain are termed configurations. The functions are assumed to be continuous. The CIES is not suited to functions with domains that are infinite but discrete.

Since the CIES is based on the PIES, the PIES is summarized here. In each iteration, the PIES algorithm selects the configuration for which random experiments shall be performed. This selection is based on the expectation

value of the entropy change due to the reevaluation. From the results of the performed Bernoulli trials, a discrete probability distribution $p_t(i)$ is calculated, where $p_t(i)$ is the probability that configuration i is the optimum. The information entropy

$$S_p = -\sum_{i=1}^{i=m} [p_t(i) \ln p_t(i)]$$

measures the amount of information concerning the location of the optimum [Eq. (3.7)]. This gives a criterion for decisions. The change of the information entropy that is expected to result from an additional Bernoulli trial for configuration i is denoted $\langle \Delta S_p \rangle^i$. For each configuration, $\langle \Delta S_p \rangle^i$ is evaluated. Then, Bernoulli trials are performed for that configuration with the largest expected entropy drop. The PIES proceeds by repeatedly calculating entropy changes and adding Bernoulli trials to the most promising configurations. This method reevaluates some configurations more often than others. It distributes the Bernoulli trials among the configurations so that it maximizes the gain of information concerning the location of the maximum. In general, the PIES is more efficient than the naive strategy which performs the same number of Bernoulli trials for each configuration. Efficiency is the ratio of gain and effort. Here, the measure of gain is the entropy drop, while the measure of effort is the number of Bernoulli trials.

The rest of this chapter is organized as follows: Section 4.2 introduces the criterion for when to add new configurations, and details the algorithm. Section 4.3 describes its implementation. Examples that illustrate how the CIES works are provided in Sec. 4.4, while section 4.5 is an application of the CIES. An outlook with ideas for further research is given in Sec. 4.6, and section 4.7 concludes the chapter.

4.2 The Information Entropy Strategy for Functions with Continuous Domain

In this section, the CIES is outlined. To find the global optimum of a function with continuous domain, it is necessary that in each region of the domain the number of evaluated configurations goes to infinity in the limit of infinite computation time. However, for given finite computation time, obviously, only a finite number of Bernoulli experiments could have been performed for a finite number of configurations. The CIES needs to meet two criteria:

1. It should use the expected entropy changes to distribute the Bernoulli trials efficiently among the configurations at any time.

2. It should expand the set of analyzed configurations in the course of the optimization so that the density of analyzed configurations approaches infinity everywhere in the domain so that, given unlimited computation time, no local optima are missed.

What is needed for the CIES in addition to the PIES is a criterion to balance the reevaluation of evaluated configurations and the evaluation of new configurations.

Let M be the set of all configurations for which Bernoulli trials have been performed. Reevaluating configurations in M on average decreases the information entropy. Adding a pristine configuration to M for which no Bernoulli trials have yet been performed increases the entropy. The strategy for adding new configurations is based on a threshold for the total entropy. New configurations are added whenever the total entropy drops below this threshold, until the total entropy is above threshold. For each new configuration a small number of Bernoulli trials is performed, where small means small compared to the total number of Bernoulli trials. If the entropy is above the threshold, some of the evaluated configurations are selected for further evaluation as outlined in the PIES. Thereby, the entropy always comes back to the threshold value in the course of the optimization.

The algorithm stops when a pre-specified contingent of Bernoulli trials has been performed. Then, the last iteration which produced a probability distribution $p_t(i)$ with an entropy below the threshold or equal to the threshold is chosen. This probability distribution is the result. Why is it necessary to choose the last iteration which features an entropy below the threshold instead of the very last iteration? In the course of the optimization, the entropy is above the threshold part of the time and below the threshold part of the time. If the last iteration happens to have an entropy that is significantly above the threshold, the corresponding distribution $p_t(i)$ contains little information. In other words, if the entropy is too high, the distribution $p_t(i)$ is not able to identify good configurations with high probability. Choosing the last iteration with an entropy below the threshold ensures that the resulting probability distributions $p_t(i)$ contains a certain amount of information. Since the entropy is the measure of information, the threshold is a lower bound to the information content of the resulting probability distribution. The CIES is detailed in pseudocode at the end of this section.

The value of the threshold is to a certain degree arbitrary. But so is the absolute value of the information entropy, because it is based on a finite configuration set. Including the infinity of configurations of which nothing is known yet would yield an infinite entropy. The choice of the threshold effectively expresses the users' preference in the number of evaluated configurations versus the number of evaluations per configuration.

In this chapter the focus is on determining at what time new configurations shall be included. The issue of which new configurations to include is postponed for future work and instead new configurations are chosen at random from the domain, although other choices are possible.

The CIES in pseudocode

Initialization
Choose $n_{\text{total}} \in \mathbf{N}$
Choose $n_{\text{inc}} \in \mathbf{N}$ with $1 \leq n_{\text{inc}} \ll n_{\text{total}}$
Choose $t_{\text{threshold}} > 0$
Choose δ with $0 \leq \delta \leq 1$
Choose one or more initial configurations
Perform n_{inc} Bernoulli trials for each initial configuration
Evaluate $p_t(i)$ for each configuration i
Evaluate S_p from the $p_t(i)$

Optimization
REPEAT
IF
($S_p < t_{\text{threshold}}$ AND $n_{\text{performed}} < n_{\text{total}}$)
THEN
REPEAT
Select a new configuration at random
Perform n_{inc} Bernoulli trials
for the new configuration
Evaluate $p_t(i)$ for each configuration i
Evaluate S_p from the $p_t(i)$
UNTIL
($S_p \geq t_{\text{threshold}}$ OR $n_{\text{performed}} \geq n_{\text{total}}$)
END IF
Evaluate $\langle \Delta S_p \rangle^i$ for each configuration i
SET $\langle \Delta S_p \rangle^{\text{best}} = \min\{\langle \Delta S_p \rangle^i\}_i$
Select all configurations i with $\langle \Delta S_p \rangle^{\text{best}} \leq \langle \Delta S_p \rangle^i \leq \delta \cdot \langle \Delta S_p \rangle^{\text{best}}$
Perform n_{inc} Bernoulli trials
for each of the selected configurations
Evaluate $p_t(i)$ for each configuration i
Evaluate S_p from the $p_t(i)$
UNTIL
$n_{\text{performed}} \geq n_{\text{total}}$

Result
Choose the last iteration which produced a probability
distribution $p_t(i)$ that features an entropy $S_p \leq t_{\text{threshold}}$
This probability distribution $p_t(i)$ is the optimization result

Notes

- The probabilities $p_t(i)$ and the entropies S_p are evaluated according to Eqs. (3.5) and (3.6) respectively.
- The probability that configuration i is the best of the evaluated configurations is $p_t(i)$.
- The expected entropy changes $\langle \Delta S_p \rangle^i$ can either be evaluated according to Eq. (3.10) or estimated according to Eq. (3.15). The estimation requires that a sufficient number of Bernoulli trials have been performed for each configuration to work properly. Therefore, n_{inc} should not be too small. In all tests, $n_{\text{inc}} \geq 100$ worked well.
- Instead of a fixed number n_{inc}, a function can be used (cf. Sec. 3.6.1).
- Refer to Sec. 3.6.1 for information about the δ.
- At any time, $n_{\text{performed}}$ is the number of performed Bernoulli trials.
- The optimization result is a probability distribution.

4.3 Implementation

The CIES was implemented with the Mathematica 6.0 computer algebra system. The implementation uses the approximation Eq. (3.15) to evaluate the $\langle \Delta S_p \rangle^i$. The version control was done with Eclipse Ganymede software. For the application described in Sec. 4.5, it was necessary to connect the implementation of the CIES with a program that performs ray tracings. LightTools 6.1.0 was chosen for that purpose. To connect the programs, a routine was written that transfers data between Mathematica and LightTools using the API functions provided by LightTools.

4.4 Illustrative Examples

4.4.1 Intended Purpose

It is the purpose of this section to illustrate how the CIES works. Two example optimizations are performed and the progresses of the optimizations are displayed with sequences of figures. Here, the optimization processes are explained qualitatively. Quantitative results are given for the application in Sec. 4.5.

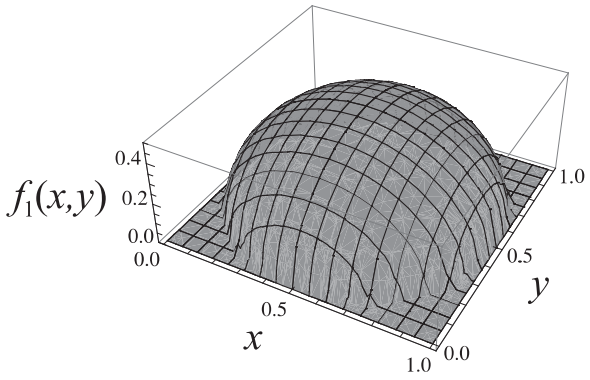

Figure 4.1: The first continuous example function: $f_1(x, y)$

4.4.2 The Example Functions

Usually, an objective function has more than two variables. In this section, the optimization results will be displayed graphically. This is possible only when the objective functions have no more than two variables. For this reason, the example functions used here depend on two variables. Two example functions have been chosen, one of them with a single maximum and one of them with two maxima. Both example functions are defined on the square domain with $x \in [0, 1]$ and $y \in [0, 1]$.

The first continuous example function is:

$$f_1(x,y) = \begin{cases} \sqrt{0.25 - (x-0.5)^2 - (y-0.5)^2} & \text{, if } ||(x,y) - (0.5,0.5)|| < 0.5 \\ 0 & \text{, otherwise.} \end{cases}$$

Its maximum is at $(x, y) = (0.5, 0.5)$ and has the value $f_1(0.5, 0.5) = 0.5$. Figure 4.1 depicts the first continuous example function.

The second continuous example function is defined as:

$$\begin{aligned} f_2(x,y) = \quad & 0.2 \ \exp\left(\frac{-(x-0.25)^2 - (y-0.25)^2}{0.08}\right) \\ + \ & 0.275 \ \exp\left(\frac{-(x-0.75)^2 - (y-0.75)^2}{0.02}\right) \\ + \ & 0.2. \end{aligned}$$

Its global maximum is at $(x, y) \approx (0.75, 0.75)$ and has the value $f_2(0.75, 0.75) \approx 0.4754$. The other maximum is at $(x, y) \approx (0.25, 0.25)$ and

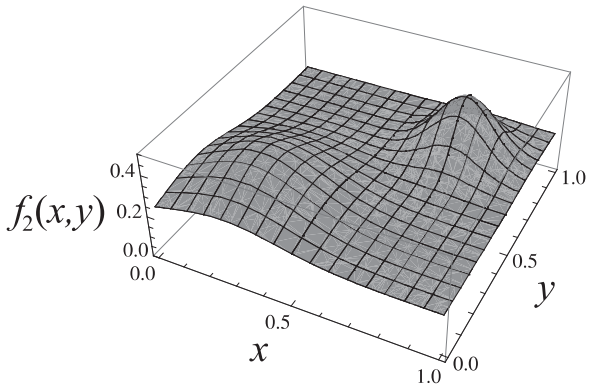

Figure 4.2: The second continuous example function: $f_2(x, y)$

has the value $f_2(0.25, 0.25) \approx 0.4$. The second continuous example function is shown in Fig. 4.2.

The CIES has no direct access to the function values. Bernoulli trials are performed as it was described for the test functions in the previous chapters: To perform a Bernoulli trial for a certain configuration (x, y), a random number between zero and one is generated. If the random number is smaller than $f(x, y)$, the result is true, otherwise it is false.

4.4.3 Settings of the Example Optimizations

Some values must be chosen as an initialization at the start of the CIES.

The settings for the optimization of the first continuous example function are as follows. The number n_{total} was set to 5×10^5. The numbers of additional Bernoulli trials were determined according to the function $n_{\text{inc}}(n_{\text{old}}) = \text{round}(\max\{100,\ n_{\text{old}}/10\})$, where n_{old} is the number of Bernoulli trials previously performed for the configuration at hand. The threshold was set to $t_{\text{threshold}} = 1.5$, and δ was set to 0.7. A single initial configuration was chosen at random: $(x, y) \approx (0.8571, 0.3594)$.

The following settings have been chosen for the optimization of the second continuous example function. This time, n_{total} was set to 4×10^5. Again, the function $n_{\text{inc}}(n_{\text{old}}) = \text{round}(\max\{100,\ n_{\text{old}}/10\})$ determined the number of additional Bernoulli trials. The threshold was set to $t_{\text{threshold}} = 2.25$, and δ was set to 0.7. Again, one initial configuration was chosen at random: $(x, y) \approx (0.9356, 0.2667)$.

4.4.4 Discussion of the Illustrative Examples

In each iteration of an optimization with the CIES, for each configuration the probability $p_t(i)$ that this configuration is the best of all evaluated configurations is evaluated. For objective functions with two variables, the status of an optimization at a certain iteration can be depicted by plotting $p_t(i)$ over the positions of the configurations. In the following figures, the data points are joined for clarity.

Figure 4.3 illustrates the progress of the optimization of the first continuous example function with a series of such plots. It shows that at the beginning of the optimization the maximum of the first example function was localized roughly, and that the region in which the maximum was located with high probability became smaller in the course of the optimization.

Figure 4.4 shows how the Bernoulli trials are distributed among the configurations at the end of this optimization. The CIES spend much more effort in the evaluation of configurations with a function value close to the maximum than with other configurations. This means that the effort was distributed very efficiently.

Figure 4.5 shows the progress of the optimization of the second continuous example function. Graph (a) shows two roughly equally high peaks, one at each maximum. Graph (b) shows a high peak at the global maximum and a small one at the other maximum. The other graphs show a peak at the global maximum only. In the beginning of the optimization, both maxima of the second continuous example function were identified. Afterwards, the CIES determined which of the maxima is the global one.

Figure 4.6 shows the distribution of the Bernoulli trials at the end of the optimization of the second continuous example function. Configurations in the vicinity of the lower maximum were evaluated more often than configurations that are far away from the maxima, and configurations close to the global maximum were evaluated much more often than all other configurations. This means that again the CIES distributed the effort with high efficiency.

During the optimization of the first continuous example function, 489 iterations were performed and 498 configurations were evaluated. The total number of Bernoulli trials performed during the optimization of the first continuous test function was 501,523.

During the optimization of the second continuous example function, 207 iterations were performed and 2184 configurations were evaluated. The number of Bernoulli trials performed in the course of the optimization of the second continuous example function was 400,203.

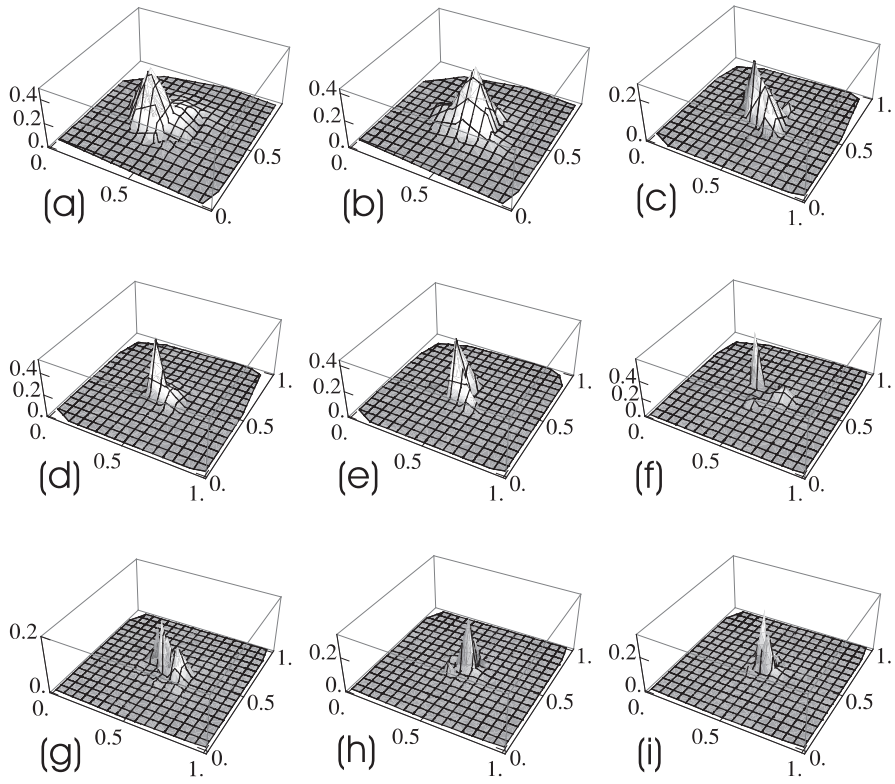

Figure 4.3: Progress of the optimization of $f_1(x, y)$. Each graph shows the status of the optimization after a certain number of iterations. The probability $p_t(i)$ is plotted over the position of the configurations. The horizontal axes represent the coordinates x and y of the configurations. The x-axis is in front, the y-axis at the right side of the plot. The vertical axis represents p_t. Graph (a) depicts the status of the optimization after the first iteration, (b) after the 61st iteration, (c) after the 121st iteration, (d) after the 181st iteration, (e) after the 241st iteration, (f) after the 301st iteration, (g) after the 361st iteration, (h) after the 421st iteration, and (i) after the 481st iteration. A total of 489 iterations were performed.

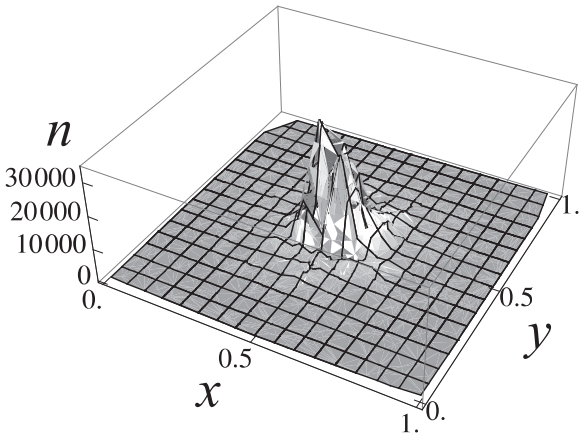

Figure 4.4: Distribution of the Bernoulli trials at the end of the optimization of $f_1(x, y)$. The number of Bernoulli trials performed for a configuration is n. This number is plotted over the position (x, y) of the configurations.

4.4.5 How does the Continuous Information Entropy Strategy work?

In the course of an optimization, the entropy varies around the threshold. This means that the number of configurations that have large probabilities $p_t(i)$ also varies around a certain value. These configurations form the peak or the peaks in the distribution $p_t(i)$ (compare Figs. 4.3 and 4.5). All other configurations have low probabilities $p_t(i)$. The density of configurations increases everywhere in the domain. This means that a peak that contains always about the same number of configurations becomes narrower in the course of the optimization. The size of the region that contains the configurations with large probabilities $p_t(i)$ approaches zero in the limit of infinite computation time. This region contains the optimum with high probability because all other configurations have low probabilities $p_t(i)$. This means that the CIES converges on the global optimum. Note that this works only for continuous merit functions. Otherwise, the configurations with large $p_t(i)$ may form no peaks around local and global optima.

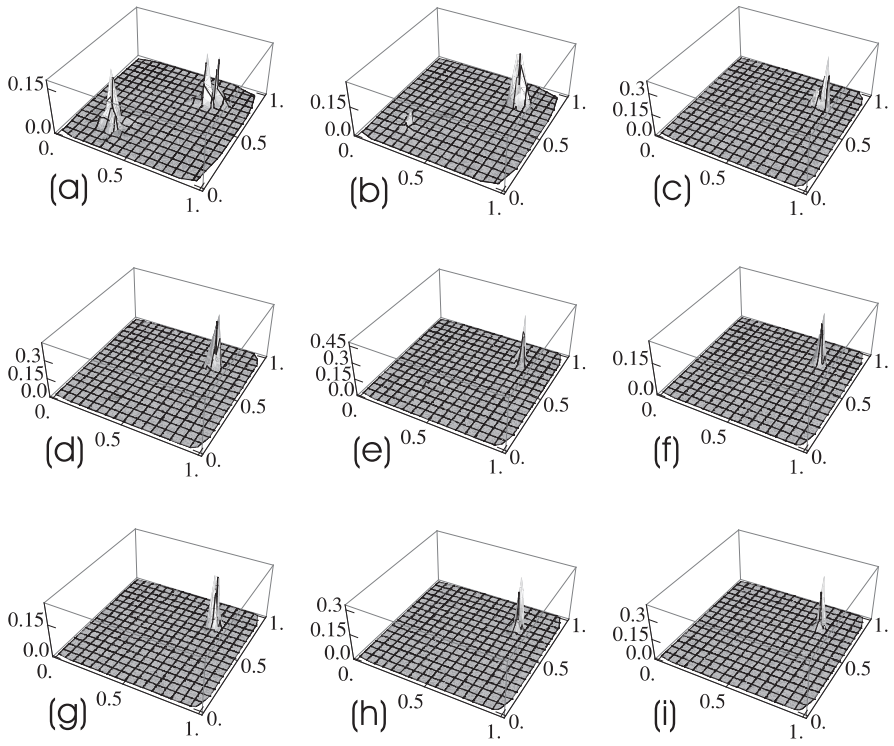

Figure 4.5: Progress of the optimization of $f_2(x, y)$. Each graph shows the status of the optimization after a certain number of iterations. The probability $p_t(i)$ is plotted over the position of the configurations. The horizontal axes represent the coordinates x and y of the configurations. The x-axis is in front, the y-axis at the right side of the plot. The vertical axis represents p_t. Graph (a) depicts the status of the optimization after the first iteration, (b) after the 26th iteration, (c) after the 51st iteration, (d) after the 76th iteration, (e) after the 101st iteration, (f) after the 126th iteration, (g) after the 151st iteration, (h) after the 176th iteration, and (i) after the 201st iteration. A total of 207 iterations were performed.

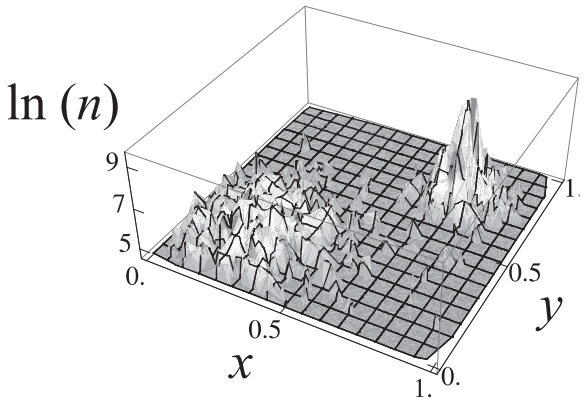

Figure 4.6: Distribution of the Bernoulli trials at the end of the optimization of $f_2(x, y)$. The number of Bernoulli trials performed for a configuration is n. The natural logarithm of n is plotted over the position (x, y) of the configurations.

4.5 Application

4.5.1 Overview

This section describes an application of the CIES. A non-imaging optical system was optimized with the CIES. It is a secondary concentrator that is part of a design of a solar concentrator. This secondary concentrator could be optimized with the CIES since its performance can be evaluated via Bernoulli experiments. The CIES determines how the Bernoulli trials are distributed among the configurations. The CIES does not perform the Bernoulli trials. Consequently, an additional program is needed that performs Bernoulli trials on demand and returns the results to the CIES. In the case of the optimization of a non-imaging optical system, a ray tracing program performs the Bernoulli trials. The next subsection explains ray tracing through virtual models.

4.5.2 Ray Tracing and Bernoulli Trials

The standard method to test optical systems is ray tracing. Today, several programs for ray tracing exist. What follows is a short description of ray tracing.

At first, a virtual model of the optical system is created. This means that all relevant surfaces, solids, material properties and surface properties of the optical system are defined in a data file. The virtual model is displayed

as a three-dimensional plot. The numbers that define shape and optical properties, e.g., curvatures of surfaces and indices of refraction, are termed parameters.

In addition to the virtual model a virtual light source and a virtual target are created. The virtual light source is defined by its geometry and the distribution of the produced light. The target is defined by a zone on a surface and an angular range.

Rays are chosen according to the distribution of the light produced by the source. The rays can be chosen pseudo-randomly, or via a quasi-random sequence (e.g., Sobol sequence). Each of the rays is traced through the virtual model. A ray follows a straight line until it hits one of the optical surfaces. Then its new direction is calculated. If the ray is reflected at the surface, its new direction is calculated using the law of reflection. If it is refracted, the new direction is calculated with Snell's law. The ray then follows a straight line until it hits one of the optical surfaces, and so on. If the ray reaches the target, its point of intersection with the target is stored together with the direction of the ray.

When a ray tracing is performed in connection with the CIES, the rays must be chosen pseudo-randomly. Each ray is then a Bernoulli trial. If the ray hits the target and its direction lies within the specified angular range, the result is true. Otherwise, the result is false.

Since the task of the CIES is not to test but to optimize an optical system, several of the parameters must be changeable. These parameters are termed optimization parameters. A configuration assigns a value to each optimization parameter. Each configuration represents one optical system.

How do the CIES and the ray tracing program cooperate? Imagine that the CIES has determined that n_{inc} Bernoulli trials shall be performed for the configuration $(x_1, x_2, x_3, x_4, x_5)$. Then the value x_1 is assigned to the first optimization parameter, x_2 to the second, and so on. Then, n_{inc} rays are traced through the optical system. The number of the rays that have reached the target and are in the specified angular range is returned to the CIES. The CIES and the ray tracing program form a client-server-system. The CIES is the client and the ray tracing program is the server.

4.5.3 Statement of Problem

In regions with much direct sunlight it is advantageous to concentrate solar radiation before it is converted into electricity because photovoltaic cells are more efficient when used with concentrated radiation. In contrast, in regions where most of the incoming solar radiation is diffuse, photovoltaic cells are used without a concentrator because diffuse radiation cannot be

concentrated.

A solar concentrator is a device that concentrates sunlight on a target. The solar radiation that enters the concentrator is redirected so that it falls onto the target. The target is smaller than the entrance aperture and the ratio of the entrance aperture area to the target area is termed geometrical concentration.

Parabolic reflectors are often used as concentrators. Parabolic reflectors produce circular focal spots. However, rectangular photovoltaic cells can be manufactured more efficiently than round ones. For this reason, a secondary concentrator that converts a circular light distribution into a quadratic one was optimized with the CIES. This secondary concentrator also enhances the concentration.

Previous work on this topic was done by Ning et al. [25]. A lot of work has been done concerning similar topics. For example, Chen et al. proposed the use of a kaleidoscope as a flux homogenizer [8]. Ries et al. analyzed sample designs of the kaleidoscope [29]. While the kaleidoscope lowers the concentration, the secondary concentrator proposed here enhances it, but produces a slightly less uniform light distribution.

4.5.4 Solution Statement and Objective Function

Figure 4.7 shows the virtual model of the secondary concentrator at the beginning of the optimization. It is a solid with a refractive index of 1.5. The light enters the device through the upper planar surface and leaves it through the lower planar surface. These two apertures are parallel to each other. The top surface is approximately circular, the bottom surface is quadratic. The height of the device is 60mm. The side surfaces redirect the light due to TIR (total internal reflection). The four side surfaces are spline patches. A spline patch is a surface that is constructed by interpolating a grid of control points. Each of the four sides is controlled by twelve points, i.e., the shape of the sides is determined by 48 degrees of freedom. The shape of the side surfaces was varied in the course of the optimization.

The secondary concentrator has symmetries. The four side surfaces have the same shape, and each of them is mirror-symmetric with respect to its midline. These symmetries reduce the number of degrees of freedom. The number of degrees of freedom is further reduced by the fact that the shapes of the upper and the lower surfaces must match the shape of the light source and the shape of the target. Five effective degrees of freedom are left, i.e., the shape of the four sides is controlled by five optimization parameters.

Figure 4.8 shows the virtual light source and the target. The circular light source is situated in the entrance aperture of the secondary concentrator. It

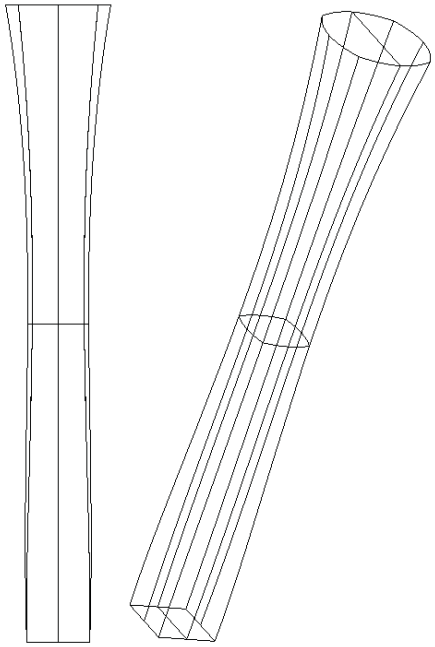

Figure 4.7: The secondary concentrator at the beginning of the optimization. Left: side view. Right: perspective view.

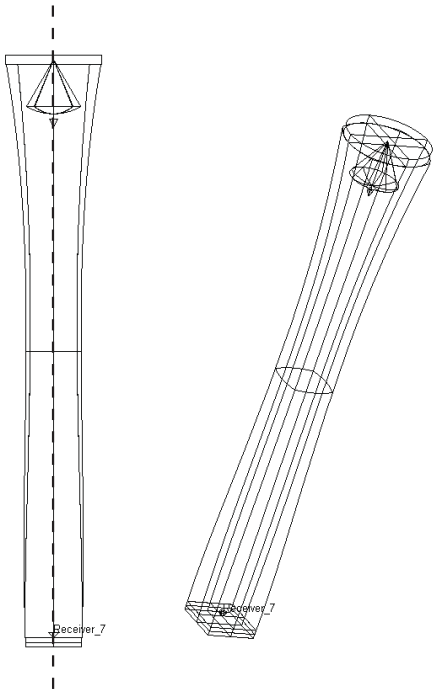

Figure 4.8: The secondary concentrator at the beginning of the optimization with virtual source and target. Left: side view. Right: perspective view. The dashed line is the optical axis.

simulates the focal spot of a paraboloidal reflector. The radius R of the source is 5 mm. Each point of the source emits light into a cone with an angle of aperture of 60 degrees, i.e., the radiation is confined to 30 degrees on each side of the optical axis. The cone is drawn in Fig. 4.8. The light rays enter the secondary concentrator immediately when leaving the source. The refraction changes the angle of aperture of the cone to 38.94 degrees. The target is located at the lower side of the secondary concentrator and is of quadratic shape with a side length L of 5.9082 mm. This value is chosen so that the light source and the target feature the same étendue. Etendue is defined in the following subsection. The target accepts all rays that are tilted less than 30 degrees with respect to the optical axis. The shape of the side surfaces is to be optimized so that as much light as possible reaches the target with an incidence angle of less than 30 degrees.

4.5.5 Etendue

The virtual light source and the target have the same étendue. What is étendue? Etendue is the most important invariant in optical engineering. To define the étendue of a bundle of rays a plane that intersects the ray bundle is used. The variables x and y are Cartesian coordinates on the plane. All the rays of the bundle that pass through an arbitrary point (x, y) on the plane form a cone. The solid angle of this cone is $\Omega(x, y)$. The projection of this solid angle onto the plane is $\Omega_p(x, y)$. The étendue is

$$\tilde{E} = n_{\text{refract}}^2 \int\int \Omega_p(x, y)\, dx\, dy,$$

where n_{refract} is the index of refraction of the material around the plane.

If $\Omega_p(x, y)$ is constant on a certain area A_{face} and zero everywhere else, then the étendue is $\tilde{E} = n_{\text{refract}}^2\ \Omega_p\ A_{\text{face}}$. The étendue does not depend on the choice of the plane. The importance of the étendue is due to the fact that the étendue of a bundle of rays does not change when the bundle passes through an optical system. For information concerning étendue refer to [40].

Now the étendue of the virtual light source is calculated. The area of the light-emitting surface is

$$A_{\text{source}} = \pi R^2 \approx 78.54 \text{ mm}^2.$$

The projected solid angle of the cone with the half-angle of 30 degrees is

$$\Omega_p = \pi sin^2(\pi/6) \approx 0.7854.$$

The half-angle of aperture is the angle between the axis of a cone and its lateral surface. This yields an étendue of

$$\tilde{E}_{\text{source}} = \Omega_p \; A_{\text{source}} \approx 61.69 \text{ mm}^2.$$

Here, n_{refract} equals one.

The size of the target was chosen so that it features the same étendue as the source. The projected solid angle of the target is again 0.7854. Together with the refractive index of 1.5 this yields:

$$\tilde{E}_{\text{target}} = (1.5)^2 \; \Omega_p \; L^2 \approx 61.69 \text{ mm}^2,$$

where $L \approx 5.9082$ mm is the side length of the quadratic target.

4.5.6 Optimization Settings

Before the optimization starts, some settings need to be specified. The five optimization parameters are x_1, x_2, x_3, x_4 and x_5. These parameters define the shape of the side surfaces of the secondary concentrator. The domain in which the algorithm chooses its new configurations for evaluation is the hypercube with $x_1 \in [-22, -18]$, $x_2 \in [-17, -13]$, $x_3 \in [-9, -5]$, $x_4 \in [-2, 2]$, and $x_5 \in [-2, 2]$. A single initial configuration was chosen, namely $(x_1, x_2, x_3, x_4, x_5) = (-22, -17, -9, -2, -2)$. Figure 4.7 shows the geometry corresponding to this initial configuration. The merit of the initial configuration is 0.79, i.e. 79% of the rays emitted by the virtual source reach the target within the specified angular range. The threshold to which the entropy is compared was set to 1.5. The value 0.8 was assigned to δ. The function $n_{\text{inc}}(n_{\text{old}}) = \text{round}(\max\{100, \; n_{\text{old}}/10\})$ was used to determine the number of additional Bernoulli trials. Here, the number of Bernoulli trials previously performed for the configuration at hand is n_{old}, whereas the number of additional Bernoulli trials is $n_{\text{old}}/10$, rounded to the nearest full number, if this number is larger than 100, and 100 otherwise. The algorithm was stopped when the number of Bernoulli trials exceeded 5×10^5.

4.5.7 Optimization Result

In the course of the optimization, 2527 configurations were evaluated. Figure 4.9 shows how the number of evaluated configurations increased in the course of the optimization. It confirms that periods of reevaluation alternated with periods in which new configurations were evaluated. Periods of pure reevaluation are visible as regions of a constant number of configurations in the plot. Periods of pure reevaluation occur when the entropy happens

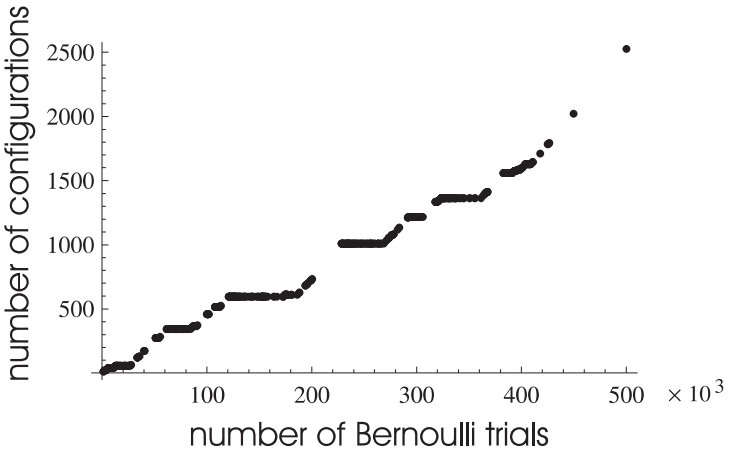

Figure 4.9: Optimization of the secondary concentrator: The number of evaluated configurations is plotted over the number of Bernoulli trials performed.

to be above the threshold for a while. A total of 638 iterations and 500,290 Bernoulli trials were performed.

During an optimization, the entropy varies around the threshold. Figure 4.10 shows the entropy S_p for each iteration.

Strictly speaking, the result of the optimization is a probability distribution. The last iteration with an entropy below 1.5 is the next to last iteration. The probability distribution $p_t(i)$ that was evaluated during the next to last iteration is the optimization result. As much as 2022 configurations have been evaluated up to this iteration. Only 50 out of the 2022 configurations of this distribution have probabilities $p_t(i) > 10^{-5}$. These probabilities are plotted in Fig. 4.11, ordered by value.

In the following, the secondary concentrator with the highest probability of being the best is examined. It is referred to as "the optimized secondary concentrator". Figure 4.12 shows this optimized secondary concentrator. It has a merit of 0.91. The configuration

$$(x_1, x_2, x_3, x_4, x_5) = (-21.37, -16.62, -7.37, -0.57, -1.72)$$

corresponds to this secondary concentrator.

To analyze the optimization result, after the optimization an additional ray tracing was performed with the optimized secondary concentrator. This time, no angular range was specified on the target. The rays where generated quasi-randomly. Almost all traced rays reached the target area (999,904 out

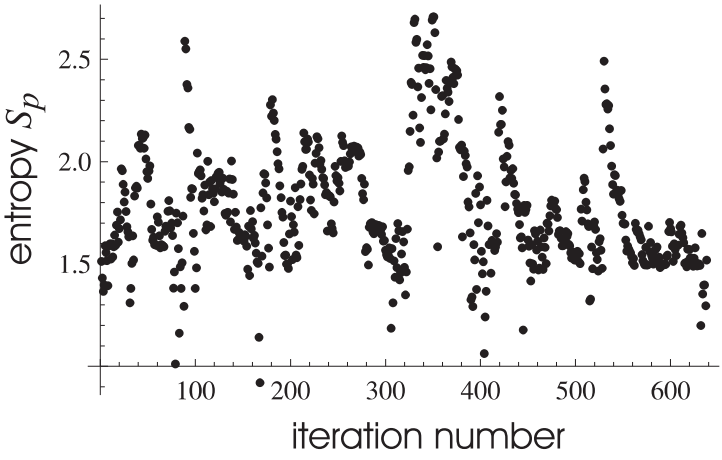

Figure 4.10: Optimization of the secondary concentrator: The entropy over the course of the optimization. For each iteration, the entropy S_p is shown.

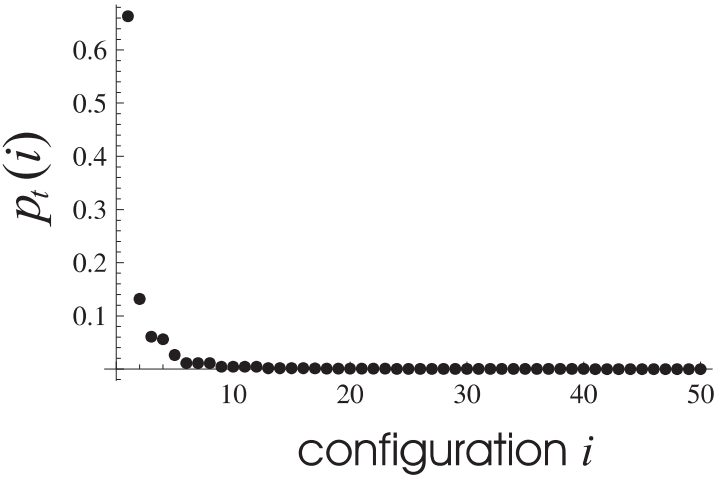

Figure 4.11: Resulting probability distribution of the optimization of the secondary concentrator: The probability that configuration number i is the best of all configurations of this distribution is $p_t(i)$. The configurations are ordered with respect to $p_t(i)$. Only 50 configurations have probabilities $p_t(i) > 10^{-5}$. The others are not shown.

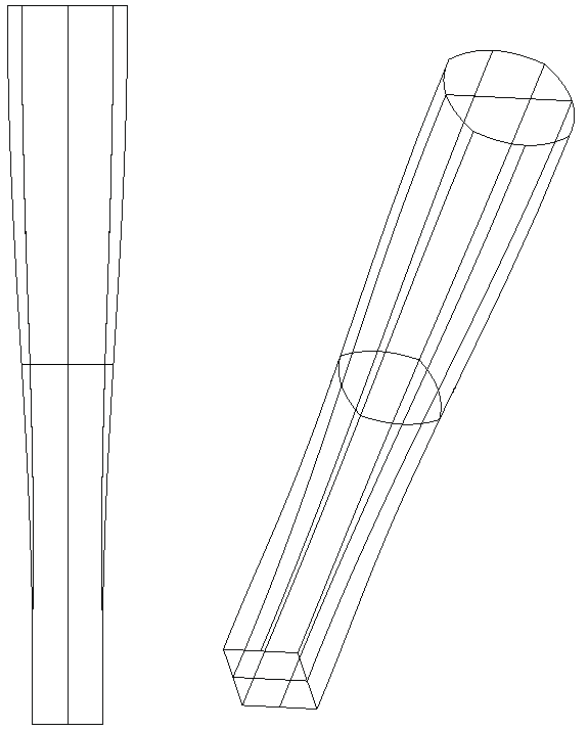

Figure 4.12: The optimized secondary concentrator.

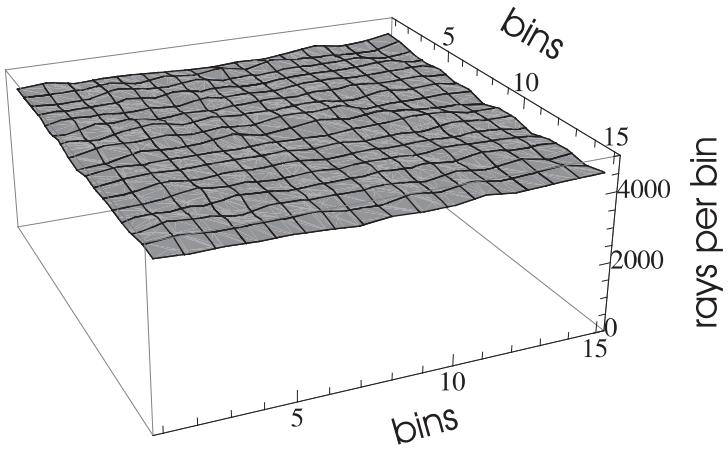

Figure 4.13: Optimized secondary concentrator: The irradiance on the target. The target is divided into 15 by 15 bins. The graph shows the number of rays received by each bin. The relative deviation of the minimal and the maximal value is 8.5%.

of 1000,000). Figure 4.13 shows the spatial distribution of the irradiance over the target. Irradiance is power per area. Since the target area is partitioned into bins of equal size, the irradiance can be measured in rays per bin. Figure 4.14 shows the angular distribution of the radiant intensity. Radiant intensity is power per solid angle.

4.5.8 Discussion of the Application

The optimized secondary concentrator has a high efficiency, since less than 10^2 out of 10^6 rays missed the target. It enhances the geometrical concentration by a factor of 2.25, which is the square of the refractive index. Figure 4.13 shows that the irradiance is almost uniform. The incidence angles of the rays reaching the target only slightly exceed 30 degrees (see Fig. 4.14). These properties are desirable, because most photovoltaic cells perform better with uniform irradiance and moderate incidence angles. Altogether, the optimized secondary concentrator produces a light distribution that is well-suited for photovoltaics.

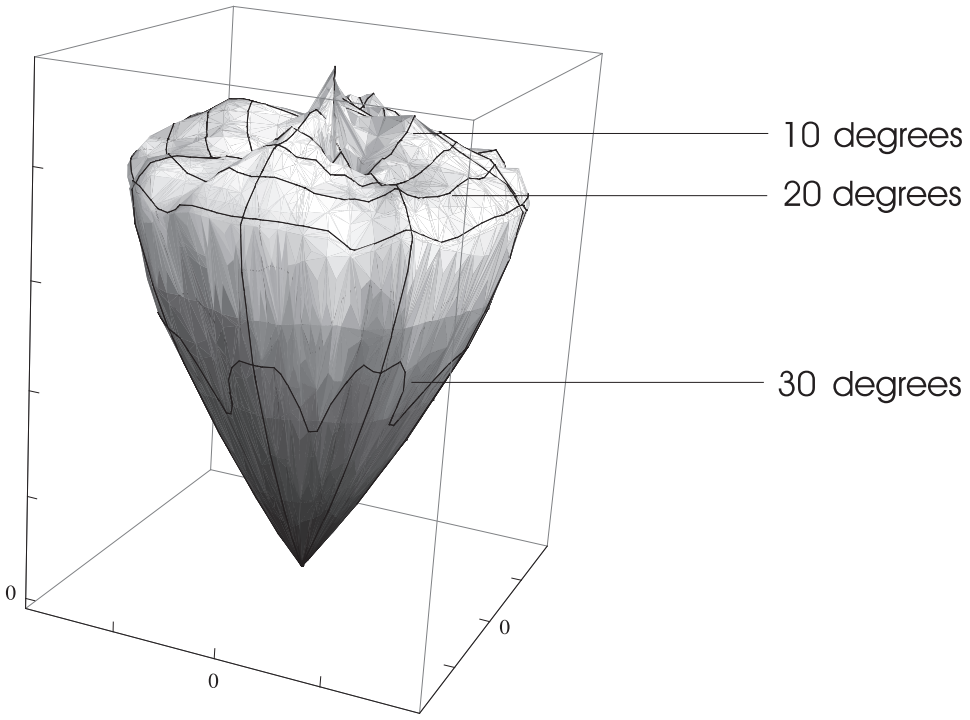

Figure 4.14: Optimized secondary concentrator: Polar plot of the radiant intensity in arbitrary units. Ideally, all rays would be confined to a cone with a half-angle of aperture of 30 degrees. The half-angle of aperture is the angle between the axis of a cone and its lateral surface.

4.6 Outlook

4.6.1 The Distribution of the Configurations

In the examples presented in Sec. 4.4 and 4.5, the new configurations were chosen at random from the domain, so they are equally distributed in it. However, to improve the efficiency of the CIES, it is desirable that the density of configurations is higher in the vicinity of the global optimum than it is in other locations. A scheme for how to choose new configurations when the entropy is below the threshold should fulfill two criteria: 1. The density of evaluated configurations should diverge to infinity everywhere in the domain. 2. Regions with higher function values should be sampled with higher density than regions with lower function values. The search for a sampling scheme that fulfils the two criteria and can be used in the CIES is an interesting field for further research.

4.6.2 How to Choose the Threshold

Up to now, no criterion for choosing the threshold $t_{\text{threshold}}$ can be given. An interesting topic for further research would be to determine how the threshold should be chosen. Maybe the information gained during the optimization can be used to adjust the threshold from time to time.

4.7 Conclusions

An information-entropy based criterion that balances the reevaluation of evaluated configurations and the evaluation of new configurations has been found. With the aid of this criterion, the PIES proposed in chapter 3 was modified so as to globally optimize stochastic functions on continuous domains. This extension is the CIES, a strategy that can optimize stochastic functions on continuous domains. The CIES roughly localizes the maximum in the beginning of the optimization. In the course of the optimization, the region in which the maximum is located with high probability is refined. The CIES spends little effort on configurations with low merit function values, thus providing high efficiency. The application shows that the CIES is capable of optimizing the design of non-imaging optical devices.

Chapter 5

Ranking and Selection with Information Entropy

5.1 Introduction

In chapters 2 to 4, methods for the optimization of stochastic functions based on information entropy were proposed. Closely related to the optimization of stochastic functions are Ranking and Selection methods. The purpose of optimization is to select an optimal element from a set. In contrast, the purpose of Ranking and Selection is to select a subset with certain optimality properties from a finite set of alternatives. A merit value, which cannot be calculated directly, but can only be estimated from the results of random experiments, is associated with each element of the set of alternatives. For example, a merit value can be the performance of a non-imaging optical system. Sections 3.2 and 4.5 explain how the performance of non-imaging optics can be estimated with random experiments. Typical questions are according to [15]:

- Which of k competing populations (or policies, or drugs, etc.) is the 'best'?
- Of the k competing populations, what are the t $(1 \leq t \leq k)$ 'best' populations with (or without) regard to order?
- Can we find a (small) subset of the k populations which contains the 'best' population?
- Can we find a subset which contains the t best populations?
- Which of the k populations are 'better' than a certain 'control' population?

In this contribution the following question is addressed: Which elements of the set of alternatives are better than a given reference value? This contribution proposes an algorithm that gains as much information as possible with a given number of random experiments concerning the question of which alternatives are better than the reference value. The algorithm is based on a criterion that makes use of the concept of information entropy. It is similar to the criterion used by the methods for the optimization of stochastic functions (chapters 2 to 4).

The rest of this chapter is organized as follows. Section 5.2 is the formulation of the problem. Section 5.3 explains how the amount of information concerning the stated problem is measured in terms of information entropy. Section 5.4 presents a criterion based on information entropy which can determine how the computational effort should be split between the alternatives at hand. In Sec. 5.5, the expected entropy changes that are needed to be able to apply the criterion are calculated for a special case. Section 5.6 presents the Ranking and Selection algorithm that uses this criterion. An empirical test of this algorithm and a comparison with a benchmark method is provided in Sec. 5.7. The last section draws conclusions.

5.2 Formulation of the Problem

The set $\{X_i\}_i$ with $i \in \{1, 2, ..., m\}$ is a set of independent random variables, representing the set of alternatives, and $r \in \mathbb{R}$ is the reference value. A parameter vector a_i is associated with each X_i. The a_i are termed configurations. Section 4.5 describes how the configurations define the geometries of optical systems. The merit value associated with configuration i is the expectation value $g_i = \mathrm{E}(X_i)$. The g_i are assumed to be unknown. In the following, an algorithm is constructed that distributes a given number of random experiments among the configurations so as to gain as much information as possible concerning the question for which i the relation $g_i > r$ holds.

5.3 Information Entropy

The amount of information is measured with the entropy. The status s_i of each configuration is:

$$s_i = \begin{cases} 1 & \text{if } g_i > r \\ 0 & \text{else} \end{cases}$$

Since g_i and r are assumed to be real numbers, the event $g_i = r$ has zero probability for all i. Here, $P_{[i]}(1)$ is the probability that $s_i = 1$, $P_{[i]}(0)$ is

the probability that $s_i = 0$, and $P_s(l_1, l_2, ..., l_m)$ denotes the probability that $s_i = l_i$ for all i. The probability P_s is a function of all l_i, and $l_i \in \{0, 1\}$ for all $i \in \{1, ..., m\}$. According to Shannon [36], the entropy of a discrete probability distribution $\{p_j\}_j$ is:

$$S = -\sum_j p_j \ln p_j.$$

The m-tuples $(l_1, l_2, ..., l_m)$ are referred to as status vectors. For example, the status vector being $(1, 0, 0, 0....)$ means that the merit value of configuration 1 is larger than the reference value, and the merit values of all other configurations are lower than the reference value.

The entropy of the probability distribution $P_s(l_1, l_2, ..., l_m)$ is:

$$S_{rs} = -\sum_{l_1=0}^{1}\sum_{l_2=0}^{1}...\sum_{l_m=0}^{1} P_s(l_1, l_2, ..., l_m) \ln P_s(l_1, l_2, ..., l_m).$$

The index rs signifies 'Ranking and Selection'. The random experiments X_i are independent. This yields:

$$P_s(l_1, l_2, ..., l_m) = \prod_{i=1}^{m} P_{[i]}(l_i).$$

The entropy of a joint distribution of independent random variables is the sum of the individual entropies:

$$S_{rs} = -\sum_{i=1}^{m}\sum_{l_i=0}^{1} P_{[i]}(l_i) \ln P_{[i]}(l_i).$$

For brevity, $P_{[i]} = P_{[i]}(0)$. The entropy can further be simplified to:

$$S_{rs} = -\sum_{i=1}^{m} \left(P_{[i]} \ln P_{[i]} + (1 - P_{[i]}) \ln(1 - P_{[i]}) \right). \tag{5.1}$$

5.4 The Criterion Based on Information Entropy

Information entropy measures the amount of information relating to the question of which alternatives are better than the reference value. An expected information gain, which is measured in terms of expected entropy change,

is associated with an additional random experiment for an arbitrary alternative. This leads to the criterion for deciding which alternative is to be reevaluated next: to gain as much information per effort as possible, each random experiment is performed for the alternative with the largest expected information gain. The expectation values of information gain have to be recalculated frequently, since they change due to the result of each random experiment.

More precisely, the probabilities $P_{[i]}$ are functions of the numbers of random experiments performed for the configurations i and of the results of these random experiments. That means that, whenever a random experiment is performed for configuration i, $P_{[i]}$ changes, and thus the entropy S_{rs} changes. The expectation value of the entropy change resulting from an additional random experiment for configuration i is $\langle \Delta S_{rs} \rangle^i$. Like the $P_{[i]}$ and S_{rs}, the $\langle \Delta S_{rs} \rangle^i$ are functions of the numbers of random experiments performed for the configurations i and of the results of these random experiments. The random experiments are performed sequentially. To gain as much information as possible, each random experiment is performed for that configuration for which $\langle \Delta S_{rs} \rangle^i$ is lowest. Note that $\langle \Delta S_{rs} \rangle^i$ changes when a random experiment is performed for configuration i. This criterion is similar to the criteria used by the IES, PIES and CIES, but for Ranking and Selection, the entropy is calculated from a different probability distribution.

5.5 Calculation of the Expected Entropy Change for the Case of Bernoulli Experiments

This section considers the special case that the random experiments are Bernoulli trials. The expected entropy change due to an additional random experiment for an arbitrary configuration i is calculated for this special case. The random experiment X_i yields the result 1 with probability g_i and the result 0 with probability $1 - g_i$. (The g_i are assumed to be unknown and can only be estimated via Bernoulli trials.) The number of Bernoulli trials performed for configuration i is n_i, and k_i is the number of occurrences of the result 1 during these n_i trials. Trials with result 1 will be called successful in the following, the others unsuccessful. The entropy S_{rs} is the entropy according to Eq. (5.1), and S_{rs}^{i+} is the entropy following a successful trial for configuration i, incrementing each n_i and k_i by one. Similarly, S_{rs}^{i-} is the entropy following an unsuccessful trial for configuration i, incrementing n_i by one and leaving k_i unchanged. The probability $\alpha_i = (k_i + 1)/(n_i + 2)$ is the probability that the next Bernoulli trial performed for configuration i will

be successful. The expected entropy change due to one additional Bernoulli trial for configuration i is:

$$\langle \Delta S_{rs} \rangle^i = \alpha_i S_{rs}^{i+} + (1 - \alpha_i) S_{rs}^{i-} - S_{rs}. \tag{5.2}$$

The entropy S_{rs} is a sum of m summands (Eq. (5.1)), where m is the number of configurations. An additional Bernoulli trial for configuration i changes only the summand number i. The other summands cancel out in Eq. (5.2). Thus, the right side of Eq. (5.2) reduces to

$$\begin{aligned} \langle \Delta S_{rs} \rangle^i = & - \alpha_i \left[P_{[i]}^{i+} \ln P_{[i]}^{i+} + (1 - P_{[i]}^{i+}) \ln(1 - P_{[i]}^{i+}) \right] \\ & - (1 - \alpha_i) \left[P_{[i]}^{i-} \ln P_{[i]}^{i-} + (1 - P_{[i]}^{i-}) \ln(1 - P_{[i]}^{i-}) \right] \\ & + \left[P_{[i]} \ln P_{[i]} + (1 - P_{[i]}) \ln(1 - P_{[i]}) \right], \end{aligned} \tag{5.3}$$

where $P_{[i]}^{i+}$ is the conditional probability of g_i being smaller than r, given that the next Bernoulli trial performed for configuration i will be successful, while $P_{[i]}^{i-}$ is the conditional probability of g_i being smaller than r, given that the next Bernoulli trial performed for configuration i will not be successful. In the case that the random experiments are Bernoulli trials, $P_{[i]}$ is the regularized incomplete beta function:

$$\begin{aligned} P_{[i]} &= I_r(1 + k_i, 1 + n_i - k_i), \\ P_{[i]}^{i+} &= I_r(2 + k_i, 1 + n_i - k_i), \\ P_{[i]}^{i-} &= I_r(1 + k_i, 2 + n_i - k_i). \end{aligned} \tag{5.4}$$

The regularized incomplete beta function is defined in Sec. 3.5.2.

5.6 The IERS Algorithm

The following algorithm utilizes the criterion for Ranking and Selection described in Sec. 5.4. The algorithm is labelled IERS for Information Entropy Ranking and Selection.

1. Specify the total number of Bernoulli experiments $n_{\text{total}} \in \mathbb{N}$.

2. Set $n_i = k_i = 0$ for all $i \in \{1, ..., m\}$.

3. Evaluate $\langle \Delta S_{rs} \rangle^i$ for all $i \in \{1, ..., m\}$ according to Eqs. (5.3) and (5.4).

4. Choose the configuration for which $\langle \Delta S_{rs} \rangle^i$ is smallest. (If this is not a unique choice, choose one one of the configurations with the smallest values of $\langle \Delta S_{rs} \rangle^i$.)
   Let i^* denote the index of the chosen configuration.
5. Perform a Bernoulli trial for the configuration i^*.
6. Increment n_{i^*}.
7. If the Bernoulli trial was successful, increment k_{i^*}.
8. Evaluate $\langle \Delta S_{rs} \rangle^{i^*}$ from the updated n_{i^*} and k_{i^*}.
9. Repeat steps 4. to 8. until n_{total} Bernoulli trials have been performed.
10. Calculate $P_{[i]}$ for all $i \in \{1, ..., m\}$ from the n_i and k_i (Eq. (5.4)).
11. Select all configurations i with $(1 - P_{[i]}) > 0.5$.

The probability $1 - P_{[i]}$ is the probability that the merit value of configuration i is larger than r.

Notes

- Contrary to the optimization methods based on entropy, the algorithm IERS recalculates only one of the expected entropy changes per iteration. This reduces the overhead.

- More than one Bernoulli experiment can be performed per iteration. (Compare IES, end of Sec. 2.2.5 and first item of Sec. 2.3.5.)

- If n_{total} is too small, wrong choices are likely. The algorithm selects the configurations that are better than r as well as possible with the specified number of Bernoulli trials.

5.7 Test of the IERS Algorithm and Comparison of the IERS with a Naive Method

5.7.1 Test of the IERS Algorithm

This section tests the algorithm by applying it to a stochastic test function. The configurations are $a_i = (i)$ with $i \in \{1, 2, 3, ..., 100\}$. The merit values

$g_i = 0.01i$ are associated with the configurations. A Bernoulli trial for configuration i is performed as follows. A random number $\tilde{x} \in [0, 1]$ is generated. If $\tilde{x} < g_i$, the result is 1, otherwise 0. This function is labelled 'test function (IERS)'.

The IERS algorithm was applied to 'test function (IERS)' ten times. The reference value was chosen to be 0.933 and $n_{\text{total}} = 2 \times 10^5$. Each of the ten test runs returned a set of probabilities (step 10 of the algorithm). Figures 5.1 and 5.2 show the result of one of the ten test runs. Figure 5.1 plots the probabilities $(1 - P_{[i]})$ of g_i being larger than the reference value r against the configuration numbers. The configurations with $g_i > r$ have probabilities close to one, the other configurations have probabilities close to zero.

This means that the algorithm correctly identified the configurations that are better than the reference value, and the probability of making wrong choices when repeating the test is very low. This corresponds to the fact that the entropy S_{rs} associated with the set of probabilities plotted in Fig. 5.1 is very low (0.0007). For each of the ten sets of probabilities returned by the test runs, the entropy was calculated. The arithmetic mean of the entropies is 0.0011.

In the following, a naive strategy which performs the same number of Bernoulli trials for each configuration is applied to the same test problem. This naive strategy serves as a benchmark. The results of the naive strategy are compared to the results of the IERS.

5.7.2 Ranking and Selection with the Naive Strategy and 2×10^5 Bernoulli Trials

The naive strategy performs the same number of Bernoulli experiments for all configurations and calculates the $P_{[i]}$ from the results. The probabilities $P_{[i]}$ of g_i being smaller than the reference value r are calculated according to Eq. (5.4). Then the configurations i with $(1 - P_{[i]}) > 0.5$ are selected.

The naive strategy was applied ten times to 'test function (IERS)'. In each test run, 2000 Bernoulli trials were performed per configuration, which yields a total of 2×10^5 Bernoulli trials per test run. This is the same number of Bernoulli trials per test run as used by the IERS in the previous subsection. The same reference value was also used (0.933).

Figure 5.3 shows the resulting probability distribution for one of the test runs with the naive strategy. In contrast to the IERS, there are configurations with probabilities $1 - P_{[i]}$ that are neither close to one nor close to zero. That means that the naive strategy with 2×10^5 Bernoulli trials could not identify

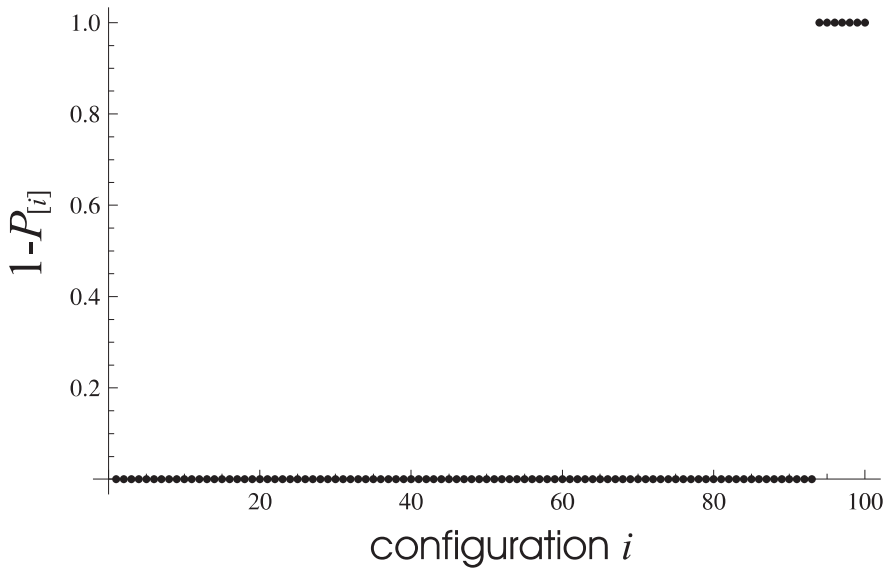

Figure 5.1: Result of the IERS. For each configuration, the probability $(1 - P_{[i]})$ of g_i being larger than the reference value is shown. The total number of Bernoulli trials is 2×10^5.

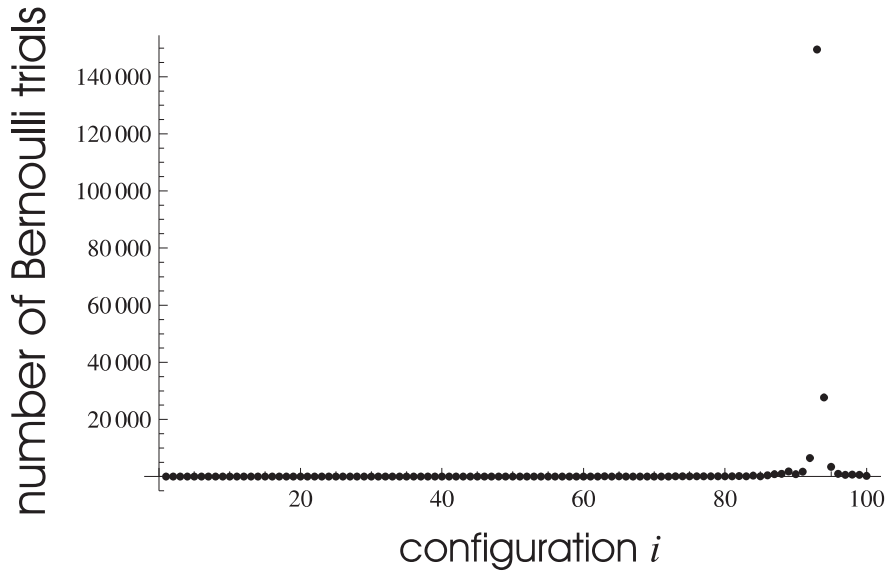

Figure 5.2: Result of the IERS. The number of Bernoulli trials performed for each configuration is shown. The total number of Bernoulli trials is 2×10^5.

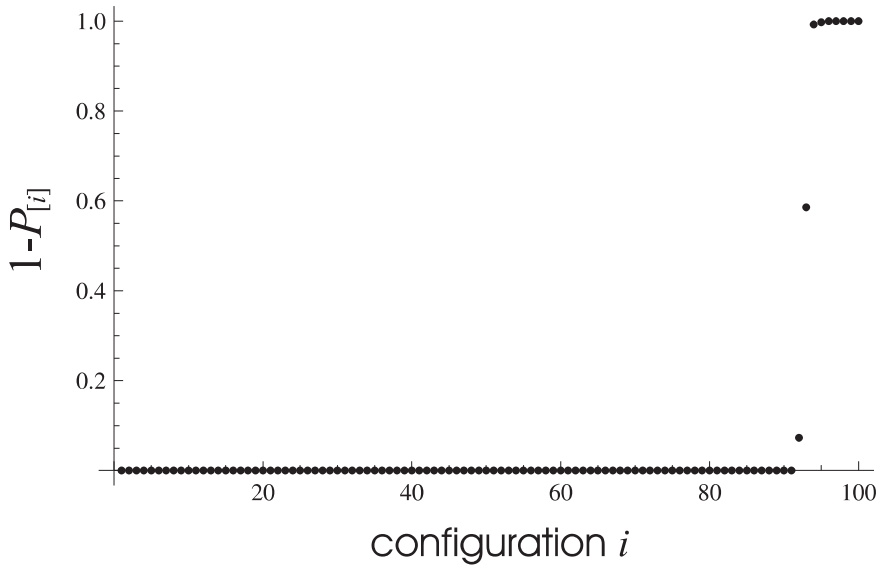

Figure 5.3: Result of the naive strategy with 2×10^5 Bernoulli trials. For each configuration, the probability $(1 - P_{[i]})$ of g_i being larger than the reference value is shown.

the configurations that are better than the reference value. In the example shown in Fig. 5.3, the probability of configuration 93 being better than the reference value is 0.5860, although its merit value $g_{93} = 0.93$ is actually smaller than the reference value. That means that the naive strategy made a wrong choice.

The information entropy is a more precise measure of the performance than the number of wrong choices. The entropy of the resulting set of probabilities $P_{[i]}$ was calculated for each of the ten test runs of the naive strategy. The average of these entropies is 0.9653. Note that this is three orders of magnitude larger than the average entropy resulting from the IERS. With the same number of Bernoulli trials, the naive strategy gained much less information than the IERS.

5.7.3 Ranking and Selection with the Naive Strategy and 10^7 Bernoulli Trials

In addition to the ten test runs with 2×10^5 Bernoulli trials, another ten test runs with 10^7 Bernoulli trials were carried out. This means that 10^5 Bernoulli trials where performed per configuration in each test run. Again, the test function was 'test function (IERS)' and the reference value was 0.933.

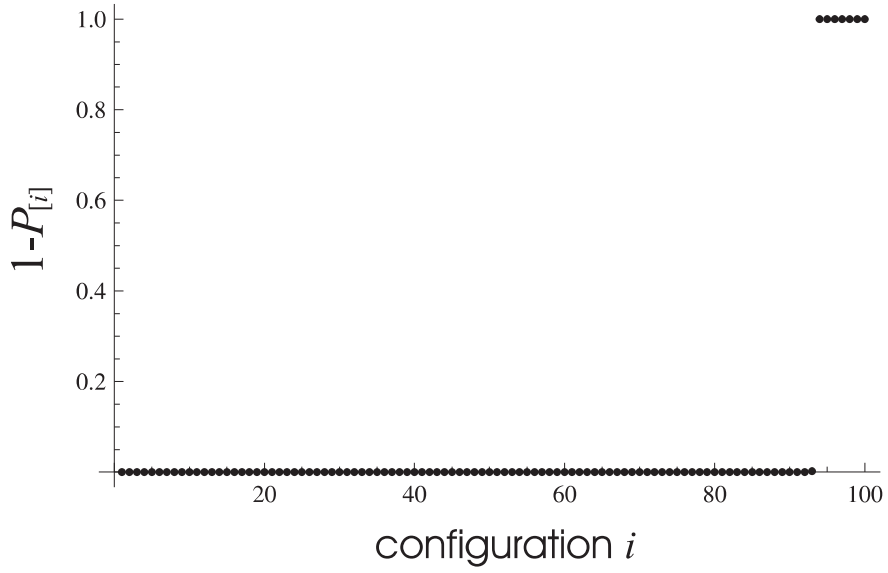

Figure 5.4: Result of the naive strategy with 10^7 Bernoulli trials. For each configuration, the probability $(1 - P_{[i]})$ of g_i being larger than the reference value is shown.

Figure 5.4 shows the resulting set of probabilities for one of the test runs with the naive strategy and 10^7 Bernoulli trials. The configurations with $g_i > r$ have probabilities close to one, the other configurations have probabilities close to zero.

This means that the naive strategy with 10^7 Bernoulli trials correctly identified the configurations that are better than the reference value, and that the probability that wrong choices occur when the test is repeated is very low. For each of the ten test runs with the naive strategy and 10^7 Bernoulli trials, the entropy of the resulting set of probabilities was calculated. The average of the entropies is 0.0065. This is larger than the average of the entropies resulting from the test runs of the IERS, but of the same order of magnitude.

5.7.4 Comparison

Table 5.1 compares the results of the three Ranking and Selection procedures. The table shows that, on average, the IERS reached a lower entropy with 2×10^5 Bernoulli trials than the naive strategy did with the same number of trials. The resulting entropy of the IERS was three orders of magnitude smaller than that of the naive strategy when both procedures used as many

Table 5.1: Comparison of three different Ranking and Selection procedures. For each of the three procedures 'IERS with 2×10^5 Bernoulli trials', 'naive strategy with 2×10^5 Bernoulli trials' and 'naive strategy with 10^7 Bernoulli trials', ten test runs were carried out and for each test run the entropy of the resulting probability distribution was calculated. For each of the three procedures, the average entropy is the arithmetic mean of the entropies of the ten test runs. The standard deviation is the standard deviation of the ten entropies. The expectation values of the entropies lie within the confidence intervals with a probability of 95%. The bottom row shows how many wrong selections occurred during the ten test runs. The expression 'wrong selection' refers to the case of a configuration with $g_i < r$ being selected as well as to the case of a configuration with $g_i > r$ being not selected. The 'naive strategy with 2×10^5 Bernoulli trials' selected a configuration with a merit value $g_i < r$ four times. The other procedures always made correct selections.

	IERS with 2×10^5 Bernoulli trials	Naive strategy with 2×10^5 Bernoulli trials	Naive strategy with 10^7 Bernoulli trials
Average entropy	0.0011	0.9653	0.0065
Standard deviation	0.0013	0.2313	0.0066
Confidence interval (95%)	$[0.0002, 0.0020]$	$[0.8000, 1.1306]$	$[0.0018, 0.0112]$
Number of wrong selections	0	4	0

as 2×10^5 Bernoulli trials. The confidence intervals of the 'IERS with 2×10^5 Bernoulli trials' and the 'naive strategy with 2×10^5 Bernoulli trials' do not overlap, so the difference is significant.

For comparison, the naive strategy was tested with 10^7 Bernoulli trials. The entropy reached by the naive strategy when using 10^7 Bernoulli trials is of the same order of magnitude as that of the IERS. The difference between the entropies of the 'IERS with 2×10^5 Bernoulli trials' and the 'naive strategy with 10^7 Bernoulli trials' is not significant.

It can be concluded that for this test problem the IERS performs significantly better than the naive strategy.

5.8 Conclusion

This chapter has shown how the information entropy can be used to best select those alternatives from a set which are better than a reference value when the effort is specified. The empirical test indicates that the algorithm presented here is superior to the naive method, which performs the same number of random experiments for each alternative.

Chapter 6

Summary

In this thesis, a connection between optimization and information theory has been developed and explored. The optimization of stochastic functions is a fast growing field of research. It is utilized in optical engineering, operations research and many other fields.

When stochastic functions are optimized, two difficulties arise additional to those that have to be faced in the optimization of deterministic functions. The first difficulty is that the number of random experiments has to be determined for each configuration. The second difficulty is how to decide when new configurations should be evaluated and when old ones should be reevaluated.

In this work, solutions to these difficulties based on the concept of information entropy were presented. Criteria for decisions using information entropy were introduced, and three methods were developed which employ these criteria for the optimization of stochastic functions. In addition, a Ranking and Selection method based on information entropy was developed.

The strategy developed in chapter 2 is termed the Information Entropy Strategy (IES). It uses the concept of information entropy to gain information about the location and value of the global optimum of a stochastic function with high efficiency. The IES is suitable for functions with finite domains. It was tested on example functions. The tests indicated that the IES is able to identify the existing local maxima and is able to determine which of them is the global maximum. Furthermore, the tests indicated that the IES performs more efficiently than the naive benchmark strategy both in terms of the number of function evaluations and in terms of time consumption.

The strategy proposed in chapter 3 is the Projection Information Entropy Strategy (PIES). It is designed to gain information concerning the location of the global optimum of a stochastic function with a finite domain with optimal efficiency. Information about the value of the optimum is not sought, but is

gained as a byproduct. The main difference to the IES is that the entropy is calculated using a projected probability distribution instead of the original distribution. The chapter includes approximations and numerical methods for keeping the computational overhead small. The PIES was also tested on example functions. The tests indicated that the PIES identifies the location of the global maximum much more efficiently than the naive strategy and the IES in terms of function evaluations. The accuracy of the approximations was shown in a separate test. In addition, the computer experiments indicated that the PIES is more efficient than the naive strategy in terms of time consumption.

Chapter 4 proposed an algorithm for the optimization of stochastic functions on continuous domains (Continuous Information Entropy Strategy or CIES). The CIES is an extension of the PIES proposed in chapter 3. The test optimizations in chapter 4 indicated that the CIES can identify the local maxima of a function on a continuous domain and can determine which of them is the global maximum. The application showed that the CIES can optimize non-imaging optical systems.

Chapter 5 showed that the information entropy can be used for 'Ranking and Selection' in a way similar to the optimization methods.

This work was not concerned with the question of which configurations should be chosen from the domain for evaluation. This means that the methods developed do not replace a method for choosing new configurations, but they can be combined with such a method. All together, the empirical data verified that the strategies succeed in solving test problems with high efficiency.

Chapter 7

Abbreviations and Symbols

7.1 Abbreviations

CIES	Continuous Information Entropy Strategy
IERS	Information Entropy Ranking and Selection
IES	Information Entropy Strategy
PIES	Projection Information Entropy Strategy
SANE	simulated annealing in noisy environments [7]
TIR	total internal reflection

7.2 Symbols

$\langle ... \rangle$	expected value
A	short name for a term used in the calculation of $\langle \Delta S \rangle^j$
A_{face}	area of a face
A_{source}	area of a light source
a_i	configurations: elements of a function domain
α_j	probability that the next Bernoulli trial for configuration j will yield a positive result

a_{opt}	position of the global optimum of the merit function at hand
B_1, B_2	short names for terms used in the calculation of $\langle \Delta S_{ap} \rangle^j$
B	Euler beta function
B_z	incomplete beta function
b_i	instruction how to perform Bernoulli trials for configuration i
c_1, c_2	constants used to determine n_{inc}
δ	threshold used to determine which evaluated configurations are reevaluated
$\langle \Delta S \rangle^j$	expected entropy change resulting from one additional Bernoulli trial for configuration j
$\langle \Delta S_{ap} \rangle^j$	approximation of $\langle \Delta S_p \rangle^j$
$\langle \Delta S_p \rangle^{\text{best}}$	minimum of the $\langle \Delta S_p \rangle^j$ i.e. expected entropy change resulting from the reevaluation of the most interesting configuration
$\langle \Delta S_p \rangle^j$	expected change of S_p resulting from one additional Bernoulli trial for configuration j
$\langle \Delta S_{rs} \rangle^i$	expected change of S_{rs} resulting from one additional Bernoulli trial for configuration i
$E(X)$	expected value of X
$\tilde{E}$	Etendue
$\tilde{E}_{\text{source}}$	Etendue of a light source
$\tilde{E}_{\text{target}}$	Etendue of a target
$\epsilon, \epsilon_{\max}$	error estimates
ε	small parameter in a series expansion
ε^{j-}	change of $p_t(j)$ resulting from an unsuccessful Bernoulli trial for configuration j
ε^{j+}	change of $p_t(j)$ resulting from an successful Bernoulli trial for configuration j

$f(x)$	arbitrary function
$f_1(x,y),\ f_2(x,y)$	example functions optimized with the CIES
$F(x)$	$F(x) := -x \ln x - (1-x)\ln(1-x)$
g	merit function value
g_i	merit function value of configuration i
$g_i^{(p1)},\ g_i^{(p2)}$	functions used to test the PIES
g_{opt}	merit function value of the global optimum of the merit function at hand
h, i, j	index variables which enumerate configurations
I_z	regularized incomplete beta function
k	number of successful Bernoulli trials
k_i	number of successful Bernoulli trials performed for configuration i
L	side length of a square target
l_i	binary digits
m	number of configurations
M	set of all evaluated configurations
max U	maximum of the set U
n	number of Bernoulli trials
N	number of points in a Monte Carlo integration
$\mathbb{N}$	the set of the natural numbers
naive strategy	a method that performs the same number of random experiments for each configuration
n_i	number of Bernoulli trials performed for configuration i
n_{inc}	number of additional Bernoulli trials
$n_{\text{performed}}$	number of performed Bernoulli trials

n_{start}	number of Bernoulli trials performed for each configuration before the optimization algorithm starts
n_{old}	number of Bernoulli trials performed before the iteration at hand
n_{refract}	refractive index
n_{total}	total number of Bernoulli trials
Ω	solid angle
Ω_p	projected solid angle
p	probability
$P_a(g)$	probability for all merit values to be below g
$P_a^{(0)}(g)$	probability for all merit values to be below g calculated before a new Bernoulli trial is carried out
$P_b(g,i)$	probability of configuration i having a merit lower than g
$P_b^{(0)}(g,i)$	probability of configuration i having a merit lower than g calculated before a new Bernoulli trial is carried out
$P_b^{j-}(g,i)$	probability of configuration i having a merit lower than g given that the next Bernoulli trial will be performed for configuration j and will yield a negative result
$P_b^{j+}(g,i)$	probability of configuration i having a merit lower than g given that the next Bernoulli trial will be performed for configuration j and will yield a positive result
$P_{\text{bin}}(n,k,g)$	Binomial distribution
$p(g,i)$	probability density of configuration i having the merit g
$P_{[i]}$	$P_{[i]} = P_{[i]}(0)$
$P_{[i]}^{i-}$	is the probability that $s_i = 0$ given that the next Bernoulli trial will be performed for configuration i and will yield a negative result
$P_{[i]}^{i+}$	is the probability that $s_i = 0$ given that the next Bernoulli trial will be performed for configuration i and will yield a positive result
$P_{[i]}(l)$	$P_{[i]}(l)$ is the probability that $s_i = l$

$p_{\mathrm{opt}}(g,i)$	probability density of configuration i being the best of all evaluated configurations and having the merit g
$P_s(l_1,l_2,...,l_m)$	$P_s(l_1,l_2,...,l_m)$ is the probability that $s_i = l_i$ for all $i \in \{1,2,...m\}$
$p_t(i)$	probability that the global maximum is at configuration i
q_1, q_2	arbitrary real numbers
r	reference value: The IERS determines which configurations have a merit larger than r
R	radius of a circular light source
$\mathbb{R}$	the set of the real numbers
round(u)	u rounded to the nearest integer
S	entropy
S_I, S_{II}	short names for terms used in the calculation of S
s_i	status of a configuration: $s_i = 1$ if $g_i > r$; $s_i = 0$ else
$\langle S \rangle^j$	expected entropy after one additional Bernoulli trial for configuration j
S_f	entropy of the probability distribution of the location and the value of the global optimum
S^{j-}	entropy after one additional unsuccessful Bernoulli trial for configuration j
S^{j+}	entropy after one additional successful Bernoulli trial for configuration j
S_p	entropy of the probability distribution of the location of the global optimum
S_{rs}	entropy of the probability distribution used in the IERS
S_{rs}^{i-}	S_{rs} after one additional unsuccessful Bernoulli trial for configuration i
S_{rs}^{i+}	S_{rs} after one additional successful Bernoulli trial for configuration i
T	temperature in simulated annealing
$T^{(0)}, T^{j-}, T^{j+}$	short names for terms used in the calculation of $\langle \Delta S \rangle^j$

t-test	statistical hypothesis test with Student's t-distribution
$t_{\text{threshold}}$	threshold used to decide if evaluated configurations are reevaluated or if new configurations are evaluated
V	volume of a domain
V^{j-}, V^{j+}	short names for terms used in the calculation of $\langle \Delta S \rangle^j$
x, y	cartesian coordinates
$x_1, ..., x_5$	optimization parameters
X_i	random variables associated with configurations
$\tilde{x}$	random number
$z, \tilde{z}$	arbitrary real numbers

Bibliography

[1] Talal M. Alkhamis and Mohamed A. Ahmed. Simulation-based optimization using simulated annealing with confidence interval. In *Proceedings of the 2004 Winter Simulation Conference*, pages 514–519, 2004.

[2] Gad Allon, Dirk P. Kroese, Tal Raviv, and Reuven Y. Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134(1):137–151, 2005.

[3] Sigrún Andradóttir. A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference*, pages 151–158, 1998.

[4] Bjarne Andresen and Jeffrey M. Gordon. Constant thermodynamic speed for minimizing entropy production in thermodynamic processes and simulated annealing. *Physical Review E*, 50:4346–4351, 1994.

[5] Dirk V. Arnold. *Noisy Optimization with Evolution Strategies*. Kluwer Academic Publishers, 2002.

[6] Robin C. Ball, Thomas M. A. Fink, and Neill E. Bowler. Stochastic annealing. *Physical Review Letters*, 91(3):030201, 2003.

[7] Jürgen Branke, Stephan Meisel, and Christian Schmidt. Simulated annealing in the presence of noise. *Journal of Heuristics*, 2007.

[8] M. M. Chen, J. B. Berkowitz-Mattuck, and P. E. Glaser. The use of a kaleidoscope to obtain uniform flux over a large area in a solar or arc imaging furnace. *Applied Optics*, 2(3):265–271, 1963.

[9] Krishna Chepuri and Tito Homem-de-Mello. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134:153–181, 2005.

[10] Gunter Dueck. New optimization heuristics The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

[11] Gunter Dueck. *Das Sintflutprinzip*. Springer-Verlag, second edition, 2006.

[12] Gunter Dueck and Tobias Scheuer. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90(1):161–175, 1990.

[13] Shu-Cherng Fang et al. *Entropy optimization and mathematical programming*. Kluwer Academic Publishers, 1997.

[14] Bennet L. Fox and George W. Heine. Probabilistic search with overrides. *The Annals of Applied Probability*, 5(4):1087–1094, 1995.

[15] David Goldsman. Ranking and selection in simulation. In *Proceedings of the 1983 Winter Simulation Conference*, pages 387–393, 1983.

[16] David Goldsman, Seong-Hee Kim, and Barry L. Nelson. Statistical selection of the best system. In *Proceedings of the 2005 Winter Simulation Conference*, pages 178–187.

[17] David Goldsman and Barry L. Nelson. Statistical screening, selection, and multiple comparison procedures in computer simulation. In *Proceedings of the 1998 Winter Simulation Conference*, pages 159–166, 1998.

[18] Tito Homem-de-Mello. Variable-sample methods and simulated annealing for discrete stochastic optimization. *Stochastic Programming E-Print Series*, 2000-4, 2000. Publisher: Humboldt-Universität zu Berlin, Institut für Mathematik.

[19] Tito Homem-de-Mello. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, 13(2):108–133, 2003.

[20] L. Jeff Hong and Barry L. Nelson. Selecting the best system when systems are revealed sequentially. *IIE Transactions*, 39:723–734, 2007.

[21] Robert Hooke and T. A. Jeeves. "Direct search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.

[22] C. Tim Kelley. *Iterative Methods for Optimization*. SIAM, 1999.

[23] Seong-Hee Kim and Barry L. Nelson. Selecting the best system: Theory and methods. In *Proceedings of the 2003 Winter Simulation Conference*, pages 101–112, 2003.

[24] Scott Kirkpatrick et al. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[25] Xiaohui Ning, Roland Winston, and Joseph O'Gallagher. Dielectric totally internally reflecting concentrators. *Applied Optics*, 26:300–305, 1987.

[26] Juta Pichitlamken and Barry L. Nelson. Selection-of-the-best procedures for optimization via simulation. In *Proceedings of the 2001 Winter Simulation Conference*, pages 401–407, 2001.

[27] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, second edition, 2002.

[28] Ingo Rechenberg. *Evolutionsstrategie '94*. Frommann-Holzboog, 1994.

[29] Harald Ries, J. M. Gordon, and Michelle Lasken. High-flux photovoltaic solar concentrators with kaleidoscope-based optical designs. *Solar Energy*, 60(1):11–16, 1997.

[30] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

[31] George Ruppeiner. Riemannian geometry in thermodynamic fluctuation theory. *Reviews of Modern Physics*, 67(3):605–659, 1995.

[32] George Ruppeiner, J.M. Pedersen, et al. Ensemble approach to simulated annealing. *Journal of Physics I*, 1:455–470, 1991.

[33] Ralf Salomon. Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

[34] Tobias Scheffer and Stefan Wrobel. A sequential sampling algorithm for a general class of utility criteria. In *International Conference on Knowledge Discovery and Data Mining*, pages 330–334, 2000.

[35] Tobias Christian Schmidt, Harald Ries, and Wolfgang Spirkl. Strategy based on information entropy for optimizing stochastic functions. *Physical Review E*, 75:021108, 2007.

[36] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, 1948.

[37] James C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[38] Wolfgang Spirkl and Harald Ries. Optimal finite-time endoreversible processes. *Physical Review E*, 52(4):3485–3489, 1995.

[39] Virginia Joanne Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Rice University, Houston, Texas, 1989.

[40] W. T. Welford and R. Winston. *High Collection Nonimaging Optics*. Academic Press, San Diego, 1989.

Acknowledgements

Foremost, I thank my advisor Prof. Dr. Harald Ries for the interesting research topic, his support and commitment, and for his advice on scientific writing. I thank Simon Junginger for helping me with the LightTools API. I would like to thank Prof. Dr. Reinhard Noack and Prof. Dr. Hans Ackermann for their constructive comments. Thanks to Hans Philipp Annen, Dr. Ling Fu, and Dr. Ralf Leutz for helpful discussions concerning optimization and possible applications of my work.

Academic Career

Name:	Tobias Christian Werner Georg Schmidt
Date of Birth:	June, 1st, 1981
Place of Birth:	Mainz, Germany
Nationality:	German

from June 2005	Research Associate, Workgroup Optics (Prof. Ries) Dept. of Physics, Philipps-University Marburg
2005	Diploma
2004-2005	Großpraktikum and diploma thesis, Workgroup Optics
2000-2005	Student of physics, Philipps-University Marburg, subsidiary subject philosophy
1999-2000	Student of physics, Ernst-Moritz-Arndt-University Greifswald